

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E  
ELETRÔNICA**

Bruna Kobay de Freitas

**RECOMPOSIÇÃO DE SERVIÇO EM REDES DE MÉDIA  
TENSÃO BASEADA EM AGENTES INTELIGENTES**

Florianópolis

2018



Bruna Kobay de Freitas

## **RECOMPOSIÇÃO DE SERVIÇO EM REDES DE MÉDIA TENSÃO BASEADA EM AGENTES INTELIGENTES**

Trabalho de conclusão de curso submetido ao Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Diego Issicaba, Ph.D

Florianópolis

2018



Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Freitas, Bruna Kobay de  
Recomposição de serviço em redes de média tensão  
baseada em agentes inteligentes / Bruna Kobay de  
Freitas ; orientador, Diego Issicaba, 2018.  
79 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro  
Tecnológico, Graduação em Engenharia Elétrica,  
Florianópolis, 2018.

Inclui referências.

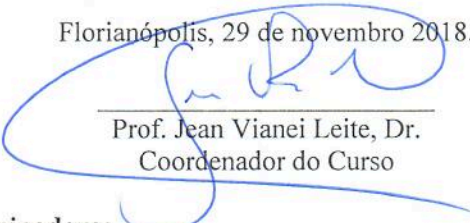
1. Engenharia Elétrica. 2. Recomposição  
automática. 3. Sistemas multiagentes. 4. PADE. I.  
Issicaba, Diego. II. Universidade Federal de Santa  
Catarina. Graduação em Engenharia Elétrica. III.  
Título.

Bruna Kobay de Freitas

## RECOMPOSIÇÃO DE SERVIÇO EM REDES DE MÉDIA TENSÃO BASEADA EM AGENTES INTELIGENTES

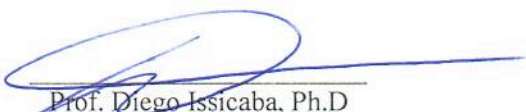
Esta Trabalho foi julgada adequada para obtenção do Título de Bacharel  
em Engenharia Elétrica/Eletrônica e aprovada em sua forma final pela  
Banca Examinadora

Florianópolis, 29 de novembro 2018.




Prof. Jean Viane Leite, Dr.  
Coordenador do Curso

**Banca Examinadora:**

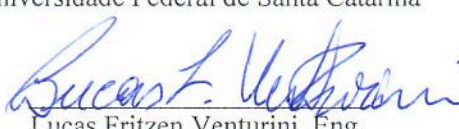


Prof. Diego Issicaba, Ph.D  
Orientador

Universidade Federal de Santa Catarina



Prof. Mauro Augusto da Rosa, Ph.D  
Universidade Federal de Santa Catarina



Lucas Fritzen Venturini, Eng.  
Universidade Federal de Santa Catarina

## RESUMO

O aumento da dependência da sociedade pela energia elétrica combinada com o avanço tecnológico dos últimos anos levaram ao aumento da complexidade dos sistemas elétricos e o surgimento do conceito de redes inteligentes. Em especial, com o crescente aumento de equipamentos sensíveis a interrupções momentâneas e pequenas variações de tensão, exige-se das concessionárias um alto grau de controle a automação da rede visando a satisfação e segurança dos consumidores. Desta forma, neste trabalho, apresenta-se uma proposta de recomposição automática de redes de média tensão utilizando sistemas multiagentes, uma técnica que vem sendo bastante aplicada para a recomposição de redes inteligentes. Para a simulação da metodologia proposta foi utilizada a plataforma *Python Agent Development* (PADE), um *framework* para desenvolvimento, execução e gerenciamento de sistemas multiagentes. Resultados de simulação ilustram a aplicabilidade da proposta para casos de estudo em rede teste.

**Palavras-chave:** Recomposição Automática. Sistemas Multiagentes. PADE.

## ABSTRACT

The increase in society's dependence on electricity combined with the technological advances of recent years has led to an increase in the complexity of electrical systems and the advent of the concept of smart grids. In particular, with the increasing of equipments that are sensitive to momentary interruptions and small variations in voltage, it is required of the concessionaires a high degree of control of the automation of the network aiming at the satisfaction and safety of the customers. Thus, in this work, a proposal is presented for the automatic recomposition of medium voltage networks using multiagent systems, a technique that has been widely applied for the realization of smart grids. For the simulation of the proposed methodology the Python Agent Development (PADE) platform was used, a framework for the development, execution and management of multiagent systems. Simulation results illustrate the applicability of the proposal to test network case studies.

**Keywords:** Automatic Recomposition. Multiagent Systems. PADE





## LISTA DE FIGURAS

Figura 1	Estrutura de classes do framework PADE no padrão UML. Fonte: (MELO, 2015) .....	36
Figura 2	Processo de identificação de um agente no PADE. Fonte: (MELO, 2015) .....	37
Figura 3	Caixa de diálogo de agentes no framework PADE Fonte: (MELO, 2015) .....	38
Figura 4	Implementação de agentes num sistema de distribuição com possibilidade de recomposição .....	41
Figura 5	Falta ocorre no ramo 3 e é isolado pela atuação da proteção B .....	41
Figura 6	Abertura das proteções a jusante da falha.....	42
Figura 7	Recomposição do sistema de distribuição após ação do agente D. ....	43
Figura 8	Sistema RBTS-BUS2.....	46
Figura 9	Sistema RBTS-BUS2 com a implementação de agentes proposta .....	47
Figura 10	Sistema RBTS-BUS2 F1 e F2 com a implementação de agentes inteligentes.....	48
Figura 11	Abertura da proteção A após ocorrência de falha entre A e B. ....	49
Figura 12	Abertura das proteções que não possuem presença de tensão em ambos os lados.....	49
Figura 13	Etapas de fechamento das chaves para recomposição do sistema após falta entre A e B.....	50
Figura 14	Diálogo entre agentes obtidos para simulação computacional de falta no ramo entre os equipamentos A e B.....	51
Figura 15	Abertura da proteção B após ocorrência de falha entre B e C. ....	52
Figura 16	Abertura das proteções que não possuem presença de tensão em ambos os lados.....	52
Figura 17	Etapas de fechamento das chaves para recomposição do sistema após falta entre B e C.....	53
Figura 18	Abertura da proteção C após ocorrência de falha entre C e D. ....	55

Figura 19 Abertura das proteções que não possuem presença de tensão em ambos os lados.....	55
Figura 20 Etapa de fechamento das chaves para recomposição do sistema após falta entre C e D.....	55
Figura 21 Abertura da proteção D após ocorrência de falha entre D e E.....	56
Figura 22 Abertura das proteções que não possuem presença de tensão em ambos os lados.....	57
Figura 23 Diálogo entre agentes obtidos para simulação computacional de falta no ramo entre os equipamentos D e E .....	57
Figura 24 Abertura da proteção F após ocorrência de falha entre F e G. ....	58
Figura 25 Abertura das proteções que não possuem presença de tensão em ambos os lados.....	58
Figura 26 Etapa de fechamento das chaves para recomposição do sistema após falta entre F e G.....	58
Figura 27 Abertura da proteção G após ocorrência de falha entre G e H. ....	59

## LISTA DE TABELAS

Tabela 1	Referências de estudos realizados com SMAs em sistemas de distribuição e os sinais utilizados para a realização das metodologias propostas. ....	31
Tabela 2	Exemplo de comunicação entre agentes.....	34



## LISTA DE ABREVIATURAS E SIGLAS

SMA	Sistemas Multi Agentes .....	20
PADE	Python Agent DEvelopment framework .....	20
GD	Geração Distribuída.....	23
JADE	Java Agent DEvelopment Framework .....	24
SCADA	Supervisory, Control and Data Acquisition .....	26
FIPA	Foundation for Intelligent Phisical Agents .....	33
IEEE	Institute of Electrical and Electronics Engineers.....	34
KQML	Knowledge Query and Manipulation Language .....	34
AMS	Agent Managemet System .....	35
DF	Directory Facilitator .....	35
MTS	Message Transport System.....	35
UML	Unified Modeling Language .....	35
AS	Agente Switch .....	39
RBTS	Roy Billinton Test System .....	45



## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>19</b>
1.1 OBJETIVOS .....	20
1.2 ESTRUTURA DO DOCUMENTO .....	20
<b>2 LEVANTAMENTO DO ESTADO DA ARTE .....</b>	<b>23</b>
2.1 MÉTODOS DE RESTAURAÇÃO EM SISTEMAS ELÉTRICOS DE DISTRIBUIÇÃO .....	23
2.2 SÍNTESE DO CAPÍTULO .....	31
<b>3 SISTEMAS MULTIAGENTES .....</b>	<b>33</b>
3.1 CONCEITO DE AGENTES .....	33
3.2 COMUNICAÇÃO ENTRE AGENTES .....	34
3.3 PLATAFORMA PARA DESENVOLVIMENTO DE SMA ...	35
3.4 SÍNTESE DO CAPÍTULO .....	37
<b>4 SISTEMA DE RECOMPOSIÇÃO AUTOMÁTICO IMPLEMENTADO .....</b>	<b>39</b>
4.1 CONSIDERAÇÕES INICIAIS .....	39
4.2 PROCESSO DE RESTAURAÇÃO PROPOSTO .....	39
4.3 SÍNTESE DO CAPÍTULO .....	43
<b>5 SIMULAÇÕES DE RESTAURAÇÃO DE REDE .....</b>	<b>45</b>
5.1 ESTUDO DAS SIMULAÇÕES REALIZADAS COM O SISTEMA RBTS .....	45
5.1.1 Falta após agente A: .....	48
5.1.2 Falta após agente B: .....	51
5.1.3 Falta após agente C: .....	54
5.1.4 Falta após agente D: .....	56
5.1.5 Falta após agente F: .....	57
5.1.6 Falta após agente G: .....	59
5.2 SÍNTESE DO CAPÍTULO .....	59
<b>6 CONSIDERAÇÕES FINAIS .....</b>	<b>61</b>
6.1 CONCLUSÕES .....	61
6.2 TRABALHOS FUTUROS .....	61
<b>REFERÊNCIAS .....</b>	<b>63</b>
<b>APÊNDICE A – Algoritmo do Agente Switch .....</b>	<b>73</b>





## 1 INTRODUÇÃO

Os principais objetivos de sistemas de energia podem ser descritos como “produzir energia elétrica e transportá-la até terminais de equipamentos em pontos de consumo” e “manter os equipamentos dentro de limites operacionais aceitáveis”. A crescente dependência da sociedade pela energia elétrica combinada com o desenvolvimento tecnológico originaram sistemas elétricos maiores e de maior complexidade. Consequentemente, problemas técnicos, ambientais e de sustentabilidade têm emergido, requerendo estratégias para a expansão do sistema de maneira a atender os objetivos de confiabilidade e qualidade. Do ponto de vista da sociedade, a fonte de energia sempre deve estar disponível, entretanto, isto não é sempre possível devido a falhas intrínsecas de equipamentos e do sistema, as quais podem levar a falta de fornecimento em regiões onde ocorrem as falhas.

Especificamente para a análise de sistemas de distribuição, falhas de fornecimento são assumidas como causada pela operação de equipamentos de proteção e interrupção de serviço. Dentre as principais causas que levam a falta de fornecimento encontram-se falhas em equipamentos, descargas elétrica, curtos-circuitos, entre outras. Além de interromper o fornecimento de energia, esses eventos, se não isolados rapidamente, podem causar danos estruturais nas linhas e equipamentos instalados.

Por muitas décadas, grande parte dos investimentos e esforços tecnológicos concentraram-se nos segmentos de geração e transmissão de energia elétrica. Entretanto, com o avançar dos anos, uma maior atenção vem sendo aplicada ao setor de distribuição. Este fato decorre em vista de que aproximadamente 90% das faltas de fornecimento de energia ao consumidor são provenientes de falhas no sistema de distribuição [1]. Deste modo, um crescente esforço vem sendo empregado neste segmento, principalmente quanto à confiabilidade da entrega do serviço ao consumidor final.

A modernização do setor de energia elétrica é um assunto que vem sendo amplamente discutido pelo mundo e que impulsionou os estudos das redes elétricas inteligentes. Junto com o grande avanço da tecnologia nos últimos anos, principalmente nas áreas de automação e comunicação, diversas pesquisas para automatizar o processo de restauração da rede estão sendo realizadas.

A restauração da rede é uma ação de emergência, que exige um planejamento cuidadoso, e pode ser entendida como uma configuração

de rede temporária até que a causa do problema seja removida ou reparada, para que o sistema possa ser retornado ao estado normal. O artigo [2] sintetiza os dois objetivos principais da restauração: (i) recuperar o máximo de serviço possível aos clientes afetados e (ii) ser implementado o mais rápido possível. Esses objetivos podem ser alcançados com a automatização e descentralização dos processo de restauração.

A grande maioria das redes de distribuição de média tensão possuem uma topologia radial, que tem por finalidade permitir a coordenação dos dispositivos de proteção de maneira econômica. Essa topologia pode ser alterada por meio da abertura ou fechamento de chaves seccionadoras presentes nas linhas, permitindo assim isolar faltas em situações de contingências e restaurar o fornecimento através de outro ramo da rede. Além disso, essa mudança também pode ser realizada para otimizar o balanço de cargas entre os alimentadores, diminuindo a carga de um alimentador sobrecarregado ao transferir sua carga para os outros.

## 1.1 OBJETIVOS

Neste contexto, este trabalho tem por objetivo apresentar uma proposta de recomposição automática de redes de média tensão utilizando sistemas multiagentes (SMAs). Como objetivos específicos, destacam-se:

- Estudar e investigar a aplicabilidade do *middleware* PADE (Python Agent DEvelopment framework) para o problema de reconfiguração automática;
- Desenvolver e implementar uma metodologia baseada em SMAs para a reconfiguração automática de redes;
- Aplicar a metodologia para casos de estudos baseados em uma rede teste.

## 1.2 ESTRUTURA DO DOCUMENTO

Este documento está organizado como segue:

- No primeiro capítulo foi apresentada uma breve introdução do problema, assim como delineados os principais objetivos do trabalho.

- No segundo capítulo uma revisão do estado da arte relacionada a recomposição de sistemas de distribuição é apresentada, destacando-se os trabalhos com a utilização de SMAs e os sinais elétricos necessários para a realização da etapa de recomposição.
- No terceiro capítulo são apresentadas definições e conceitos de SMAs, como também, a estrutura e arquitetura da plataforma de desenvolvimento utilizada neste trabalho, o PADE.
- No quarto capítulo são discutidos os aspectos necessários para a realização da simulação, definindo os dados iniciais que um agente deve ter em sua base de dados e as lógicas necessárias para a restauração da rede.
- No quinto capítulo são apresentadas as possibilidades de reconfiguração e os resultados das simulações realizadas em uma rede teste.
- No sexto e último capítulo são descritas as conclusões do presente trabalho e as considerações finais desse estudo.



## 2 LEVANTAMENTO DO ESTADO DA ARTE

Neste capítulo apresenta-se a pesquisa do estado da arte relacionada a recomposição de sistemas de distribuição, com foco nos sinais elétricos sensorizados para o desenvolvimento desses sistemas. Visto que grande parte das faltas observadas pelos consumidores encontram-se no segmento da distribuição, espera-se que a revisão indique que o aumento da inteligência e autonomia embarcadas nos equipamentos da rede possibilite melhorias quanto ao tempo de fornecimento, melhorando os índices de confiabilidade e qualidade de energia e diminuindo custos relacionados a equipes de manutenção.

### 2.1 MÉTODOS DE RESTAURAÇÃO EM SISTEMAS ELÉTRICOS DE DISTRIBUIÇÃO

Vários sistemas para restauração de redes de energia já foram desenvolvidos. A maioria dos sistemas desenvolvidos nas décadas passadas para resolver esses problemas eram sistemas de computação centralizados, com tempo de resposta alto se comparado às novas tecnologias disponíveis. Sistemas baseados em agentes, tópico deste trabalho, compreende um novo campo de pesquisa na área de inteligência artificial, motivo de grande interesse dos pesquisadores devido a sua capacidade de agir autonomamente.

No trabalho proposto em [3], os autores têm por objetivo realizar simulações de metodologia de multi agentes proposta para a restauração da maior parte possível do sistema após a ocorrência de curto-circuito. Para o método mostrado, fontes distribuídas nos alimentadores são consideradas. Os autores ressaltam que a comunicação entre os agentes deverá ser a mesma para que erros de interpretação não sejam cometidos e que os agentes deverão ter o conhecimento da mesma perspectiva do ambiente no qual se encontram. A metodologia é simulada em um sistema que possui chaves para restauração do sistema e uma geração distribuída (GDs) próxima ao término do circuito. A partir dos resultados obtidos, os autores citam que o controle descentralizado apresenta melhores resultados em comparação ao controle centralizado, o qual podem melhorar os índices de confiabilidade do sistema.

Em [4], os autores apresentam uma metodologia multi agente para diminuir o tempo de restauração de serviço aos consumidores após desligamentos. Os autores citam que os controles de sistemas de dis-

tribuição sempre foram realizados de maneira centralizada, entretanto devido ao incremento de novos equipamentos e GDs, necessita-se aprimorar os tempos de restabelecimento de forma otimizada. Para a metodologia proposta, apenas sinais de corrente ao longo das linhas são necessários para verificar se as cargas desligadas poderão ser restauradas por outro ramo ou alimentador. Cada agente armazena somente informações topológicas locais e quando chamados se comunicam com seus vizinhos para adquirir as informações necessárias para a tomada de decisão sobre o serviço de restauração. Após uma falta, o agente líder realiza um plano de restauração conforme informações que recebeu dos demais agentes dos sistemas para verificar quais equipamentos deverão comutar com intuito de atender a maior parcela de carga possível com as restrições de operação devido à falta. O método foi implementado na plataforma JADE (*Java Agent DEvelopment Framework*). Ao término, os autores concluem que o método proposto é capaz de identificar planos otimizados para sistemas de distribuição.

No trabalho descrito em [5], tem-se por objetivo apresentar um método multiagente para a automação de sistemas de distribuição. Neste artigo são descritas as características principais que um agente pode ter. Cita-se também que para um sistema multiagente, necessita-se que haja sociabilidade entre os agentes, ou seja, que haja troca de informações entre equipamentos com intuito de obter informações relevantes do ambiente onde estes se encontram. Na metodologia proposta, cada agente pergunta ao seu vizinho se ele presenciou a falta. A partir das respostas obtidas o ramo em falta pode ser isolado e o planejamento da restauração pode ser realizado. Para a validação da solução proposta, um sistema teste com um alimentador foi utilizado. A partir dos resultados, os autores concluem que a utilização de um sistema multi agente diminui o tempo para restabelecimento de energia, o qual aumenta a confiabilidade e qualidade de energia do sistema

Em [6] os autores projetam e simulam um sistema com agentes cooperativos, com apenas representação local do ambiente do sistema. A estrutura e as operações dos comportamentos dos agentes permite que os agentes exibam uma funcionalidade totalmente assíncrona e descoordenada. Assim, uma distribuição completa de decisões e ações foram alcançadas, o que favorece o desempenho em detrimento de procedimentos de gerenciamento e restauração de energia. Neste trabalho, os autores ressaltam que a restauração mostra-se um problema para a distribuição e uma grande área de pesquisa a ser explorada. O gerenciamento de SD centralizados e com vários equipamentos distintos pode apresentar esforço computacional elevado. Como decorrência dos

resultados da simulação, os agentes conseguiram explorar ao máximo as fontes alternativas de energia para restaurar a energia de tantas subestações secundárias quanto possível.

Como os demais trabalho de multi agentes aqui descritos, em [7], os autores tem por objetivo apresentar uma estratégia para restauração de sistemas após a ocorrência de um curto circuito. No método proposto as cargas podem ser restauradas por ordem de prioridade. Para as simulações utilizou-se o ambiente de simulação JADE e considerou-se tensão constante em todos os nós. Além das chaves dos sistemas, as cargas possuem também chaves para viabilizar ordens de prioridade. Ao término, conclui-se que o método permite restauração do serviço total ou parcialmente dependendo da importância dada a cada ponto de consumo do sistema.

O trabalho apresentado em [8] mostra uma arquitetura de agentes para a restauração de sistemas de potências em ambientes de micro-redes. Cada agente foi modelado com objetivos específicos com um pequeno conjunto de restrições. Todos eles tentam otimizar seu próprio objetivo no menor tempo possível, implementando um algoritmo que escolhe primeiro quem tem a prioridade de carga mais alta e só depois distribuindo para o restante da rede.

Em [9], um protocolo de comunicação para aprimorar a organização dos agentes após a falta é apresentado. Como em [4] e [7], os autores de [9] utilizam o JADE como plataforma para a simulação do sistema de agentes. Para a proposta descrita, agentes podem comunicar se possuem ou a não a possibilidade de auxiliar na restauração de ramos desconectados após a falta. Como conclusão, os autores citam que a utilização de sistemas multiagentes em sistemas de distribuição possibilita a solução não centralizada do problema, diminuindo tempo no qual os consumidores ficam sem energia.

Em [10], apresenta-se um método multi agente híbrido que utiliza as vantagens do controle centralizado e descentralizado. Como descrito em [6], os autores citam que o controle centralizado de grandes sistemas possuem custos elevados (computacionais, operacionais, etc). Para a proposta, a isolamento da falta é responsabilidade dos agentes conectados ao longo do sistema e o planejamento da restauração é de responsabilidade do agente conectado ao alimentador. Nesta metodologia, inicia-se o processo através da realimentação de parte do alimentador onde a falta ocorre até o ramal isolado e posteriormente analisa-se com os demais agentes do sistema a possibilidade de restauração de cargas desconectadas. São utilizados algoritmos de otimização para o planejamento da restauração do serviço. As simulações são re-



alizadas no software MATLAB. Na implementação da metodologia, os agentes possuem conhecimento de sinais de fasores de tensão e corrente ao longo do sistema. Os resultados da simulação mostram que a abordagem de restauração proposta pode restaurar a energia para a zona saudável o mais rápido possível, mantendo a amplitude da tensão dentro das limitações.

Para o estudo apresentado em [11], um método de sistema de multi agentes para sistemas de distribuição com restauração é apresentado. Nesta metodologia necessita-se de sinais de sincrofasores de corrente, ou seja, fasores de corrente sincronizados no tempo (geralmente por meio de sinal GPS). A partir do momento no qual um dos agentes percebe a presença de uma corrente muito elevada (acima de gatilho estabelecido), este passará a mensagem a seus vizinhos até que um agente sinaliza que seu gatilho não foi atingido. Deste modo, isola-se a falta e inicia-se o processo de restauração do sistema. Para a simulação da proposta, um sistema de distribuição com múltiplos cenários de falta foi simulado. A partir dos resultados apresentados, observa-se que o método proposto é capaz de isolar a zona que se encontra em falha rapidamente e restaurar o sistema para as zonas não afetadas. Outro estudo com a mesma metodologia pode ser encontrado em [12].

Analogamente a [11], o estudo apresentado em [13] tem por objetivo apresentar um algoritmo multi agente para localização e isolamento de falta e restauração de sistemas de distribuição com a utilização de sincrofasores de corrente. Entretanto, diferentemente de [11], os gatilhos utilizados para estabelecer ramos em falta são a diferença angular entre dois pontos de medição. A partir desta metodologia torna-se possível diferenciar os tipos de falta presentes no sistema.

Em [14], é apresentado um sistema de multi agentes para restauração de sistemas de distribuição. Para a proposta descrita os agentes possuem comunicação com o sistema SCADA (*Supervisory, Control and Data Acquisition*) da subestação e a recomposição é realizada de maneira centralizada. Para as simulações foi utilizada a plataforma de simulação JADE. Apenas o sinal de corrente do sistema é utilizado nos agentes ao longo do alimentador. Além de simulações com proteções coordenadas e seletivas, foram também apresentados resultados sem coordenação de equipamentos de proteção. Ao término os autores citam que sistemas multi agentes mostram-se como uma solução eficiente de restauração de sistema onde prioridades e restrições podem ser consideradas.

A Tabela 1 apresenta as referências de utilização de multi agentes em sistemas de distribuição. Nesta tabela são apresentados os sinais

necessários para a utilização e as e as funcionalidades das metodologias descritas.

Referências	Requisitos (sinais)	Funcionalidade
[3]	Tensão e corrente	Restauração de serviço durante faltas
[4]	Corrente	Planejamento de restauração
[5]	Tensão e corrente	Eficiência da restauração do sistema
[6]	Tensão e corrente	Aprimoramento de sistema de distribuição quanto a proteção e restauração.
[7]	Tensão e corrente	Restauração de sistemas de distribuição
[8]	Tensão e corrente	Restauração rápida do sistema
[9]	Tensão e corrente	Restauração de sistemas de distribuição com utilização de protocolo de comunicação proposto
[10]	Fasor de tensão e corrente	Localização e isolamento da falta e restauração rápida de serviço
[11]	Sincrofasor de corrente	Proteção e restauração de sistemas
[12]	Tensão e corrente	Localização e isolamento de faltas e recomposição do serviço
[14]	Corrente	Restauração automática de sistemas de distribuição
[15]	Tensão e corrente	Localização e isolamento do local da falta
[16]	Tensão e corrente	Controle de microrredes com GDs
[17]	Corrente	Deteção de falta
[18]	Tensão e corrente	Descrever um método de gerenciamento autônomo de micro redes
[19]	Tensão e corrente	Controle descentralizado de tensão
[20]	Tensão e corrente	Controle de microrredes

[21]	Tensão e corrente	Controle de fluxo de potência em tempo real em MT com grande incidência de GDs
[22]	Tensão e corrente	Restauração de serviço
[23]	Tensão e corrente	Operação e controle de micro-redes
[24]	Tensão e corrente	Estimação de estados de sistemas de distribuição
[25]	Tensão e corrente	Conceito de controle de redes de distribuição
[26]	Tensão, corrente e frequência.	Isolamento da falta e restauração de serviço
[27]	Tensão e corrente	Otimização de controle de sistema de distribuição com GDs provendo suporte de tensão
[28]	Corrente	Isolamento de falta em sistemas de distribuição com inserção de GDs
[29]	Tensão e corrente	Controle de tensão
[30]	Tensão e corrente	Deteção de falta e restauração do sistema
		Redução de perdas
[31]	Tensão e corrente	Permitir mercado de energia entre sistemas
[32]	Tensão e corrente	Permite mudanças na proteção, controle, regras e infraestrutura da rede
[33]	Tensão, corrente e frequência.	Operações em ilhamentos com grande incidência de GDs
[34]	Tensão, corrente e frequência.	Avaliação de adequação e segurança de sistema de distribuição possibilitando ilhamento. Análise de regime permanente e dinâmico de geradores

[35]	Tensão, corrente e frequência.	Avaliação de adequação e segurança de sistema de distribuição possibilitando ilhamento. Análise de regime permanente e dinâmico de geradores.
[36]	Tensão e corrente	Isolamento de falta e restauração do serviço
[37]	Tensão e corrente	Restauração rápida do sistema
[38]	Tensão e corrente	Aplicações de self-healing em sistemas de distribuição
[39]	Tensão e corrente	Estratégia para reparo eficiente de faltas
[40]	Tensão e corrente	Restauração de sistema de distribuição com GDs
[41]	Tensão e corrente	Deteção e isolamento da falta e Restauração do sistema com GDs
[42]	Tensão e corrente	Deteção e isolamento da falta e realização de estratégia para restauração do sistema com GDs
[43]	Tensão e corrente	Reconfiguração do sistema de distribuição com diminuição das perdas
[44]	Corrente	Reconfiguração do sistema de distribuição com multi agentes (SMA) baseado em teoria dos jogos
[45]	Tensão	Restauração de serviço
[46]	—	Estado da arte de SMA utilizados para restauração de sistemas de distribuição
[47]	Tensão e Corrente	Isolação de falta e restauração do sistema de distribuição com geradores locais
[48]	Tensão	Monitoramento de tensão e falhas

[13]	Sincrofasor de corrente	Localização e isolamento de falta e restauração de sistemas
[49]	Tensão e corrente	Restauração de sistemas suportados por micro redes
[50]	————	Overview de operações descentralizadas
[51]	Tensão e corrente	Restauração de sistemas de distribuição com geradores distribuídos
[52]	Tensão e corrente	Reconfiguração de sistemas de distribuição por meio da minimização das perdas no sistema de distribuição
[53]	Tensão e corrente	Comparação entre SMA centralizados e descentralizados para restauração de serviço.
[54]	Tensão e corrente	Diminuir perdas em smart grids. Por meio de reconfiguração de sistemas.
[55]		Overview de Self-healing em smart-grids
[56]	Tensão	Controle de sistemas
[57]	————	Overview de SMA em smart grids
[58]	Tensão e corrente	Supervisão e controle de smart grids
[59]	Tensão e corrente	Restauração de sistema com possíveis ilhamentos com GDs
[60]	Tensão e corrente	Controle de tensão por meio de GDs
[61]	Tensão e Corrente	Controle de demanda em transformadores de distribuição
[62]	Tensão e corrente	Localizando e isolamento de falta e restauração de serviço
[63]	Tensão e correntes	Self-healing em smart grids
[64]	Corrente	Metodologia para identificação de topologia de redes em SD

[65]	_____	Estado da arte de localização e isolamento da falta e restauração do serviço em sistemas de distribuição
------	-------	--

Tabela 1 – Referências de estudos realizados com SMAs em sistemas de distribuição e os sinais utilizados para a realização das metodologias propostas.

2.2 SÍNTESE DO CAPÍTULO

Apesar de sistemas multi agentes ser um campo de pesquisa relativamente novo na área de inteligência artificial, a partir da análise do estado da arte realizado neste capítulo, nota-se que muitas pesquisas de restauração de redes que utilizam o conceito de SMA estão sendo desenvolvidas na área acadêmica. Observa-se que grande parte dos métodos de sistemas de multi agentes utilizam sinais de tensão e corrente, os quais possibilitam cálculos de fluxo de potência no principais pontos dos sistemas, sendo utilizada em muitas propostas. Indica-se também que a aplicação de inteligência e autonomia embarcadas em equipamentos da rede pode melhor quanto ao tempo de fornecimento, melhorando os índices de confiabilidade e qualidade de energia e diminuindo custo de equipes de manutenção do sistema.



### 3 SISTEMAS MULTIAGENTES

São apresentados neste capítulo definições e padrões da arquitetura de agentes inteligentes e de SMA, dando ênfase a plataforma de desenvolvimento PADE, ferramenta estudada neste trabalho, e sua estrutura baseada nos comportamentos definidos pela *Foundation for Intelligent Physical Agents* (FIPA) .

#### 3.1 CONCEITO DE AGENTES

Pesquisadores ainda não encontraram uma definição inteiramente aceita da palavra “Agente”, embora exista um consenso geral onde se define um agente como uma entidade autônoma flexível capaz de perceber o meio ambiente através dos sensores conectados a ele.

De acordo com *Wooldridge, 2009* [66]:

Sistemas multiagentes são sistemas compostos por múltiplos elementos computacionais chamados de agentes, que interagem entre si. Agentes são sistemas computacionais com duas importantes capacidades. Primeiro, são capazes de realizar ações autonomamente, decidindo por si só, o que precisam fazer para atingir os objetivos para que foram especificados. Segundo, agentes são capazes de interagir com outros agentes, não apenas trocando informações, mas interagindo entre si, apresentando comportamentos semelhantes às interações sociais cotidianas dos seres-humanos: cooperação, coordenação, negociação, entre outras.

Algumas características importantes que diferenciam um agente inteligente de controladores simples são as seguintes:

- **Autonomia:** refere-se a capacidade de um agente escolher suas ações independentemente, sem intervenção externa de outros agentes na rede (caso de sistemas multiagente) ou interferência humana, adaptando-se a quaisquer mudanças no ambiente dinâmico.
- **Capacidade de Resposta:** a capacidade de perceber mudanças no meio ambiente e responder a elas em tempo hábil para realizar



as ações necessárias para que seus objetivos especificados possam ser alcançados

- Comportamento Social: mesmo que a decisão do agente deva ser autônoma, ele ainda deve ser capaz de interagir com as fontes externas quando for necessário obter um objetivo específico. Também deve ser capaz de compartilhar esse conhecimento e ajudar outros agentes a resolver um problema específico.

### 3.2 COMUNICAÇÃO ENTRE AGENTES

A comunicação entre agentes é um dos componentes cruciais do SMA. É necessário que todos os agentes participantes se entendam, sendo que para isto uma terminologia comum deve ser definida.

Com o intuito de padronizar o desenvolvimento de SMA interoperáveis, o *Institute of Electrical and Electronics Engineers* (IEEE) criou, em 2002 a *Foundation for Intelligent Physical Agents*. A FIPA define padrões relacionados a três questões centrais em SMA: comunicação entre agentes, gerenciamento de agentes, e arquitetura dos agentes.

A linguagem FIPA possui características próximas a KQML (*Knowledge Query and Manipulation Language*), primeira linguagem formalizada de comunicação entre agentes, se diferenciando nas performativas e na lista de semânticas. O objetivo principal da criação da FIPA foi simplificar e racionalizar a lista de ações performativas o máximo possível e endereçar as questões de semântica, que se caracteriza como um dos principais problemas da comunicação entre agentes.

Em geral, a ação de comunicação é definida por uma performativa (por exemplo, requisitar, perguntar, informar) e um conteúdo de proposição (e.g. “a chave está fechada”). Deste modo, pares diferentes de performativas e conteúdo compõem diferentes atos de comunicação. Um exemplo de possíveis pares de ação/conteúdo aplicados a chaves de sistemas de distribuição podem ser visto na Tabela 2.

Conteúdo	Performativa	Discurso
“A chave está fechada”	Requisitar	“Feche a chave”
“A chave está fechada”	Perguntar	“A chave está fechada?”
“A chave está fechada”	Informar	“A chave está fechada”

Tabela 2 – Exemplo de comunicação entre agentes.

### 3.3 PLATAFORMA PARA DESENVOLVIMENTO DE SMA

A FIPA [67] define um modelo padronizado de plataforma de agentes que é composto de quatro elementos principais:

- Agentes: São os agentes propriamente ditos, que executam suas atividades na plataforma e que constituem o corpo do SMA.
- Agent Management System (AMS): É o agente que supervisiona o acesso e a utilização da plataforma de agentes. Só um AMS é permitido em uma plataforma multiagente. Cada agente precisa da permissão do AMS para poder entrar na plataforma e ficar visível para os demais agentes;
- Directory Facilitator (DF): Este agente provê o serviço de páginas amarelas à plataforma, ou seja, é por meio do DF que os agentes que possuem serviços para disponibilizar aos demais cadastram seus serviços e é também por meio do DF que os agentes que precisam de serviços procuram por eles.
- Message Transport System (MTS): É a camada de software que controla todas as trocas de mensagens no SMA, incluindo mensagens entre agentes numa mesma plataforma e também entre agentes em diferentes plataformas.

Ao longo dos anos, inúmeros plataformas para construção de SMA foram desenvolvidas, como o JADE, JASON e o PADE, entre outras. Neste trabalho, explorou-se a plataforma PADE, visando estudar suas funcionalidades e descobrir suas limitações.

O PADE caracteriza-se como uma plataforma desenvolvida em Python com base no padrão FIPA por (MELO, 2015) [68], mas seu criador não se detém aos seus detalhes mais específicos, buscando uma abordagem de comunicação mais direta. Um dos pontos que se diferenciava dos padrões é a arquitetura de gerenciamento de sistemas multiagentes do PADE, diferentemente do padrão FIPA, ele não contempla o agente facilitador (DF), módulo que provê o serviço de páginas amarelas para outros agentes, e não centraliza as ações do AMS no contêiner principal. No PADE, o AMS exerce funções de controle e supervisão e mantém uma tabela que contém os identificadores dos agentes. A Figura 1 apresenta a estrutura de classes do PADE no padrão UML (*Unified Modeling Language*).

Em PADE, o AMS é considerado também um agente, e necessita-se iniciá-lo primeiramente, visto a necessidade de cadastro dos demais

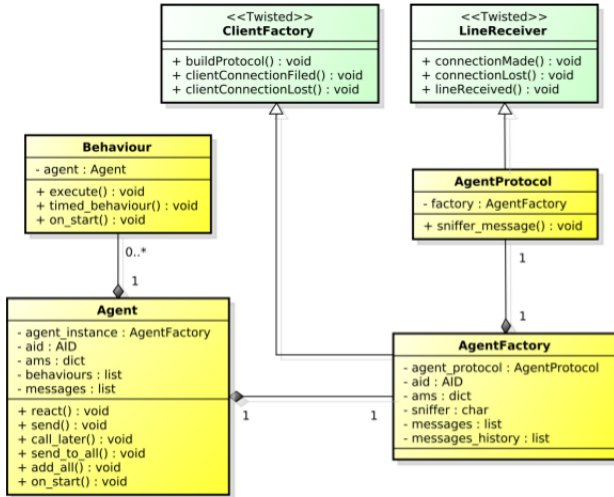


Figura 1 – Estrutura de classes do framework PADE no padrão UML. Fonte: (MELO, 2015)

agentes. Todo agente lançado na plataforma PADE primeiro precisa se “apresentar” ao agente AMS. Esse processo de identificação junto ao AMS pode ser representado por meio do diagrama de atividades apresentado na figura 2. Todo novo agente que deseje entrar na plataforma primeiramente deve enviar uma mensagem ao agente AMS informando seu identificador, o AMS então verifica em sua base de dados se não há nenhum outro agente com o mesmo identificador. Se já existir é recusado o cadastro do novo agente que terá que enviar a solicitação novamente com um novo identificador. Se não existir nenhum outro agente com o mesmo identificador na tabela do AMS, a solicitação de cadastro é aceita e a informação sobre o novo agente é enviada aos demais agentes na plataforma, assim como a tabela com os identificadores dos outros agente é enviada ao novo agente.

O Framework PADE possui também a função de um agente *Sniffer* cujo principal objetivo é enviar mensagens periódicas a agentes para realizar testes de conexão. Caso um agente não responda a mensagem, a informação é repassada ao AMS, que utilizará as informações para atualizar a tabela de agentes. Outra função realizada pelo agente *Sniffer* é a de guardar os registros das conversas entre os agentes. Um exemplo da interface gráfica do *Sniffer* pode ser visto na Figura 3,

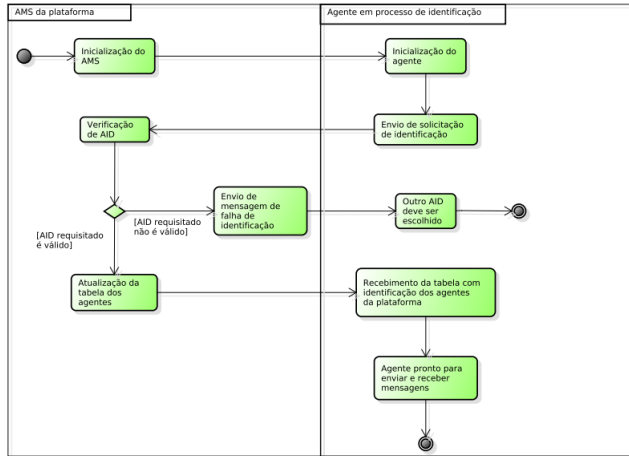


Figura 2 – Processo de identificação de um agente no PADE. Fonte: (MELO, 2015)

onde é apresentada a caixa de diálogo de agentes do framework PADE.

Por meio da identificação e listagem de cada agente da rede, permite-se visualizar o estado de cada participante do SMA durante as simulações e com o agente *Sniffer*, acompanhar as etapas que culminam na recomposição do sistema após a ocorrência de eventos e abertura de equipamentos de proteção

### 3.4 SÍNTESE DO CAPÍTULO

Neste capítulo foi exposto brevemente os conceitos que definem agentes inteligentes e os padrões definidos para estes sistemas pela FIPA. Além disso, foi apresentada a arquitetura da plataforma de desenvolvimento escolhida para este trabalho, o PADE, um framework desenvolvido em Python, uma linguagem de programação moderna e emergente.

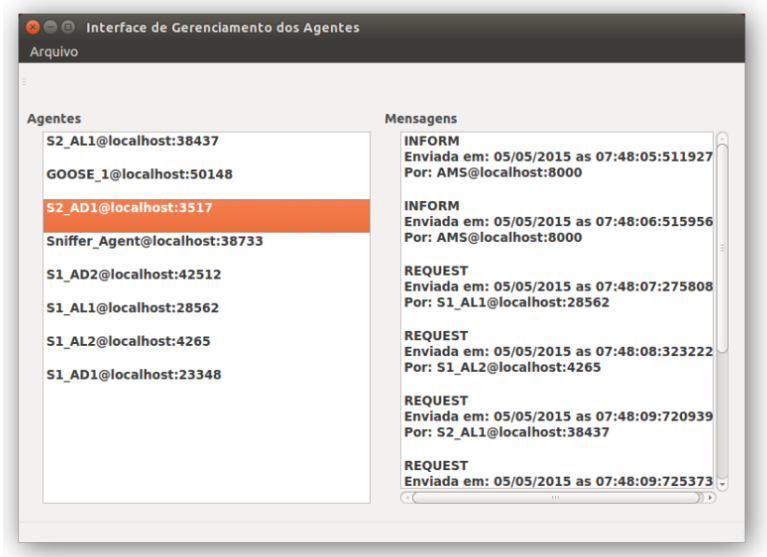


Figura 3 – Caixa de diálogo de agentes no framework PADE Fonte: (MELO, 2015)

## 4 SISTEMA DE RECOMPOSIÇÃO AUTOMÁTICO IMPLEMENTADO

Umas das soluções para gerir a crescente complexidade da operação de sistemas de distribuição é por meio da imersão de inteligência distribuída na rede. Além da possibilidade de descentralizar o controle, diminuindo o tempo para a tomada de decisões, a utilização de agentes autônomos auxilia na localização e isolamento da falta bem como na recomposição do serviço, considerando-se limites de carregamento do alimentador com interferência mínima de operadores. Neste contexto, este capítulo tem por objetivo apresentar a proposta de recomposição automática de serviço após faltas na rede.

### 4.1 CONSIDERAÇÕES INICIAIS

Para a representação da rede elétrica de sistemas de distribuição, assumiu-se que todos os pontos notáveis do sistema foram equipados com interruptores trifásicos, sendo todos estes embarcados com agentes inteligentes. Após a ocorrência de uma falta na rede, as chaves interruptoras permitem a abertura tripolar, de modo que todas as fases sejam desconectadas ao mesmo tempo. Além dos equipamentos de proteção ao longo da rede, todos os alimentadores possuem também agentes que trocam informações com outros agentes.

A partir de uma abordagem de comunicação ponto a ponto, proposta para a recomposição automática de sistemas de distribuição de média tensão, estabeleceu-se a modelagem de agentes distribuídos ao longo da rede, denominados Agent Switch (AS), sendo o código do agente apresentado no apêndice A. Neste modelo de comunicação, cada equipamento interruptor instalado na rede de média tensão possui um Agente Switch responsável por monitorar as mudanças nas correntes elétricas que percorrem as linhas, verificar a presença de tensão nos terminais onde se encontra instalado e iniciar o processo de restauração de serviço em caso de necessidade.

### 4.2 PROCESSO DE RESTAURAÇÃO PROPOSTO

Durante a operação normal do sistema, os agentes recebem informações de correntes da rede a partir de sensores locais, responsáveis

pelas percepções dos agentes, com o fim de atualizar os valores de carregamento no setor à jusante do equipamento. Abaixo são listados os dados iniciais de cada agente e alteração das crenças do mesmo a medida que informações são recebidas dos sensores de corrente e de presença de tensão.

- **ID (Identificador):** Cada agente inteligente presente na rede possui um ID único que será utilizado para identificação no envio de mensagens aos demais agentes do sistema. Este identificador é um objeto estruturado com conjunto de informações correspondentes tanto ao agente quanto ao equipamento ao qual está locado.
- **Lista de vizinhos:** Cada agente possui uma lista com os vizinhos à montante e à jusante, sendo esta lista alterável via AMS. Ressalta-se que cada agente pode ter mais de dois agentes vizinhos, nos casos onde tem-se mais de uma ramificação no mesmo nó.
- **Status da chave:** Como os agentes são instalados nos equipamentos de proteção, que por sua vez funcionam também como chaves de manobra e ditam as conexões do sistema, cada equipamento possuirá uma variável inicial que indica se a chave está aberta ( $s = 0$ ) ou fechada ( $s = 1$ ). Este parâmetro de entrada pode ser alterado via AMS. A título de exemplo, na Figura 4, os equipamentos A, B, C, E e F possuirão status de chave fechada ( $s = 1$ ) e o equipamento D possuirá status de chave aberta ( $s = 0$ ). Outros parâmetros cujos valores dependem da operação em tempo real também podem estar inclusos na estrutura de objeto chave.
- **Carregamento:** Cada agente possuirá um limite de carregamento que será utilizado para a etapa de restauração de serviço.
- **Aguarda Ocorrências na rede:** No estado normal de operação, o sistema não apresenta nenhuma ação para que objetivos de recomposição sejam atingidos. Entretanto, na ocorrência de falta e abertura de proteções, a base de crenças dos agentes possibilita a estes tomar ações para reconfiguração e atender o maior número possível de cargas.

Caso a corrente de falta ultrapasse o valor dado pela curva tempo  $x$  corrente do equipamento religador, está abrindo os contatos interrompendo o fluxo de corrente para o ponto da falta e cargas à jusante

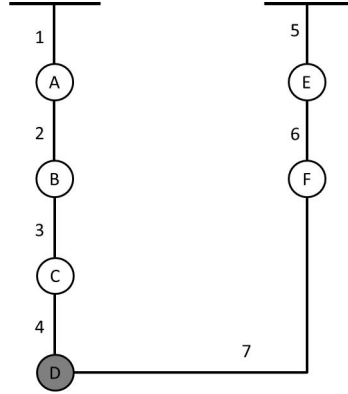


Figura 4 – Implementação de agentes num sistema de distribuição com possibilidade de recomposição

deste. Inicia-se neste ponto o processo de restauração por meio dos agentes embarcados nos equipamentos. Para os estudos aqui apresentados, assumiu-se a hipótese de proteções coordenadas, sendo o primeiro equipamento à montante da falta sempre o primeiro a atuar após a ocorrência do evento. Utilizando o mesmo sistema apresentado na Figura 4, caso uma falta ocorra no ramo 3 do sistema, a proteção B irá atuar, como mostrado na Figura 5.

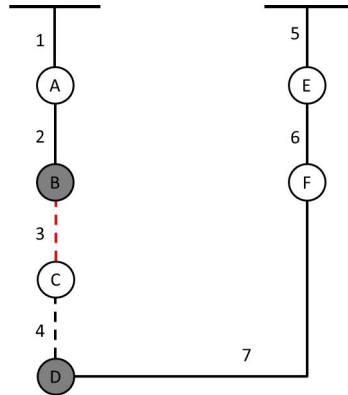


Figura 5 – Falta ocorre no ramo 3 e é isolado pela atuação da proteção B

O processo de restauração a partir da comunicação ponto a ponto entre os agentes segue os seguintes passos:



1. Proteção atua isolando a falta (com mostrado na Figura 5)
2. Agente Switch do equipamento que isolou a falta confirma a mudança de estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ).
3. Agente Switch que não identifica presença de tensão em ambos os lados do circuito, também muda o status da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ).
4. Agente Switch que abriu conexão devido à falta envia mensagem à montante e à jusante informando que isolou a falta.
  - (a) Se agente possui tensão; nenhuma ação é realizada.
  - (b) Se agente não possui presença de tensão, informa aos seus vizinhos à jusante que a falta está isolada solicitando restauração (Figura 6).
5. Se o Agente Switch possui presença de tensão a jusante e recebe solicitação de restauração, a chave encontrada realizará o fechamento do dispositivo caso este atenda requisitos de carregamento. Para o exemplo da Figura 6, a restauração ocorre pelo fechamento da chave D, como mostrado na Figura 7.

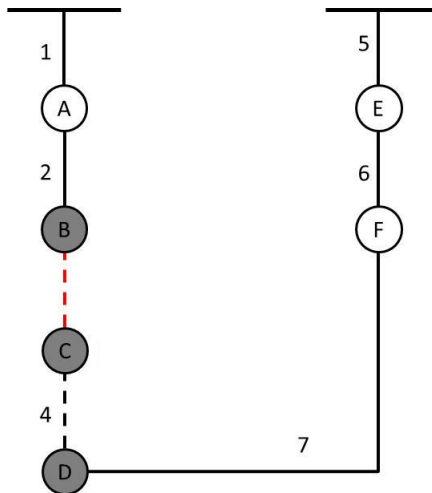


Figura 6 – Abertura das proteções a jusante da falha.

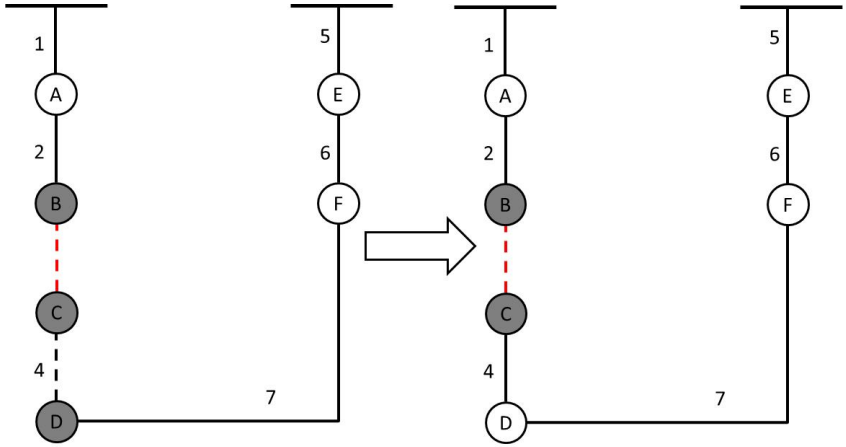


Figura 7 – Recomposição do sistema de distribuição após ação do agente D.

Após o reparo da falha, os agentes podem retornar aos estados originais do estado anteriormente a falta a partir da permissão de uma entidade superior.

#### 4.3 SÍNTESE DO CAPÍTULO

Neste capítulo apresentou-se a proposta de recomposição automática de serviço após faltas na rede, assumindo-se proteções coordenadas, detalhando algumas das etapas que serão implementadas nos agentes.



## 5 SIMULAÇÕES DE RESTAURAÇÃO DE REDE

Neste capítulo são apresentadas as possibilidades de reconfiguração de sistemas por meio da utilização de inteligência distribuída em redes de distribuição. Os estudos foram realizados utilizando o sistema RBTS (*Roy Billinton Test System*). Em todas as simulações, utilizou-se o modelo monofásico equivalente de redes elétricas, de modo que todas as aberturas e fechamentos de chaves de manobra e equipamentos de proteção foram considerados tripolares. Limites de carregamento foram desconsiderados para maximizar o número de chaveamentos nos testes.

### 5.1 ESTUDO DAS SIMULAÇÕES REALIZADAS COM O SISTEMA RBTS

O sistema RBTS foi originalmente proposto para sistemas de geração e transmissão e extensamente utilizado em para fins científicos e validação de modelagens [69]. Entretanto, grande parte das faltas de fornecimento de energia ao consumidor final, advém de falhas nos sistemas de distribuição. Deste modo, [70] propõe a extensão do sistema RBTS de transmissão para contemplar também aspectos da distribuição de energia com seus elementos de rede correspondentes.

Originalmente composto por seis barras (cinco delas com a presença de carga), o sistema RBTS é expandido para sistemas de distribuição em duas destas, conhecidas como RBTS-BUS2 e RBTS-BUS4. A barra 2 apresenta quatro alimentadores: três com consumidores residenciais e um com consumidores industriais. Já a barra 4 apresenta sete alimentadores: cinco residenciais/comerciais e três industriais. A Figura 8 apresenta o RBTS-BUS2, utilizado neste trabalho para verificação das possibilidades de recomposição de redes de distribuição de média tensão.

O alimentador da barra 2 possui tensões de 33 kV e 11 kV. As cargas residenciais e comerciais foram alocadas na baixa tensão, ou seja, sempre após os transformadores rebaixadores de 11 kV e 415 V. Consumidores com demanda de 1 MVW ou maior foram alocados na média tensão. O número total de consumidores é de 1908 e a carga média é de 12,291 MW. Nestes alimentadores nenhum banco de capacitor é previsto e as cargas são apenas descritas por suas potências ativas.

Como pode ser observado na Figura 8, o RBTS-BUS2 possui

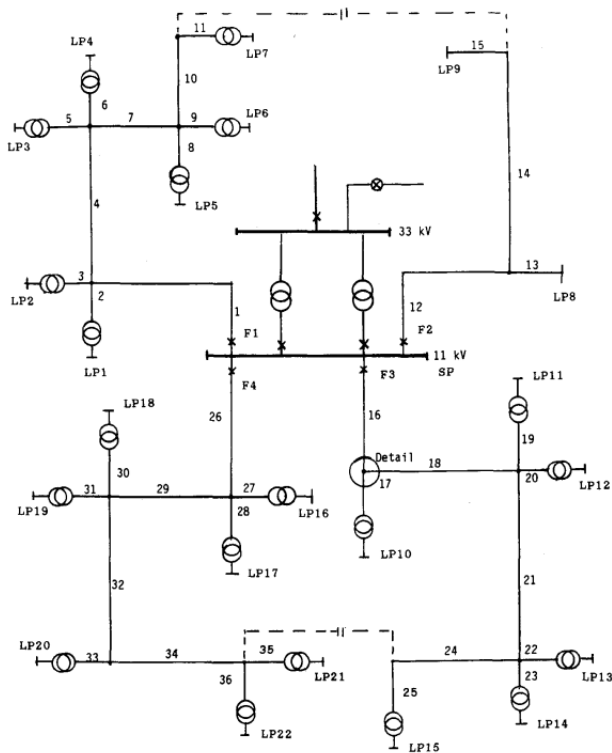


Figura 8 – Sistema RBTS-BUS2

duas chaves de manobra entre alimentadores: uma possibilitando a conexão dos alimentadores 1 e 2 e outra entre os alimentadores 3 e 4. Essas chaves possibilitam a recomposição de cargas no sistema após faltas e aberturas de proteções, de modo que possam ser utilizadas na avaliação das propostas de recomposição de sistema de maneira eficiente e simples com a utilização de agentes ao longo da rede.

Para a simulação de recomposição do sistema, foram considerados ao menos um dispositivo embarcado de agente para a proteção de cada ponto da rede com chave seccionalizadora, como apresentado na Figura 9. Note que originalmente essas chaves não são apresentadas no diagrama unifilar do sistema RBTS. Ao todo, considerou-se dezoito agentes, sendo cinco para o alimentador 1 (F1), três para o alimentador 2 (F2), cinco para o alimentador 3 (F3) e cinco para o alimentador

4 (F4). Como pode ser visto na Figura 9, cada alimentador possui agentes que além da função de manobra, também tem como função proteger as cargas e ramos do sistema. Optou-se pela implementação de um agente de chaveamento em cada alimentador a título de avaliar a proposta em seu limite.

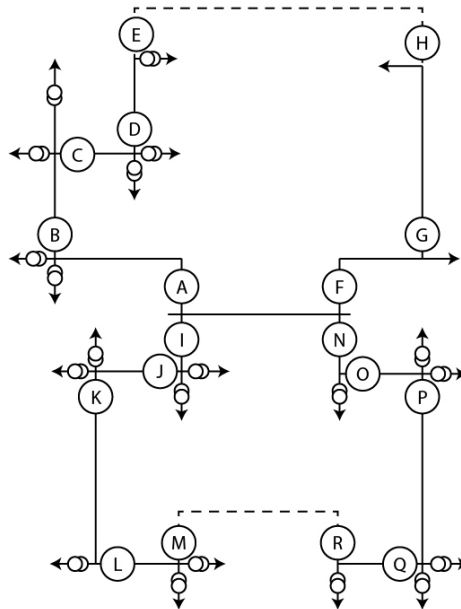


Figura 9 – Sistema RBTS-BUS2 com a implementação de agentes proposta

Em virtude da grande variedade de possibilidades de recomposição da rede, serão apenas aqui apresentados os casos para os alimentadores F1 e F2 do sistema RBTS-BUS2. Entretanto, metodologias análogas podem ser aplicadas também aos alimentadores F3 e F4. Um diagrama simplificado dos alimentadores F1 e F2 do RBTS-BUS2 com a inserção dos agentes pode ser observado na Figura 10.

Abaixo são apresentadas as possibilidades de recomposição após a ocorrência de faltas no sistema em todos os ramos do sistema apresentado na Figura 10. Em cada possibilidade, serão descritos as tomadas

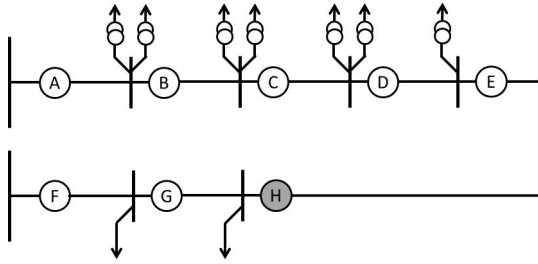


Figura 10 – Sistema RBTS-BUS2 F1 e F2 com a implementação de agentes inteligentes

de decisão dos agentes na possibilidade de comunicação ponto a ponto. Nas figuras apresentadas a seguir, círculos brancos representam proteções com chaves fechadas e círculos cinzas representam proteções com chaves abertas. Linhas contínuas são ramos onde há presença de tensão, linhas descontínuas vermelhas indicam o ramo onde ocorreu a falha e linhas descontínuas pretas representam ausência de tensão devido a chaves abertas.

### 5.1.1 Falta após agente A:

O passo a passo para a restauração do sistema utilizando a comunicação ponto a ponto após falta no ramo entre os equipamentos A e B segue abaixo:

1. Proteção A atua isolando a falta (Figura 11)
2. Agente Switch A do equipamento que isolou a falta confirma estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ).
3. Agente Switch B, C, D e E mudam estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ), ao detectar a falta de presença de tensão por um tempo determinado (Figura 12).
4. Agente Switch A envia mensagem avisando que isolou a falta ao agente B.
5. Agente Switch B mantém sua chave aberta até ordem de entidade superior. Ademais informa ao agente C que falta foi isolada. Essa informação é passada a jusante até agente H.

6. Agentes C, D e E solicitam a jusante restauração de serviço informando carregamento pré falta do setor.
7. Agente H detecta presença de tensão em apenas um dos lados e realiza fechamento de chave perante atendimento de limite de carregamento adicional ao alimentador. Este processo se repete pelas ações dos agentes E, D e C, nesta ordem. O processo de restauração é apresentado na Figura 13.
8. Fim de processo

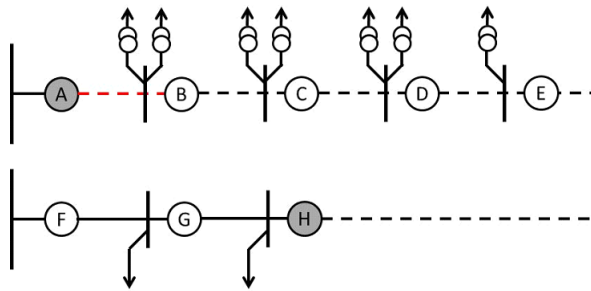


Figura 11 – Abertura da proteção A após ocorrência de falha entre A e B.

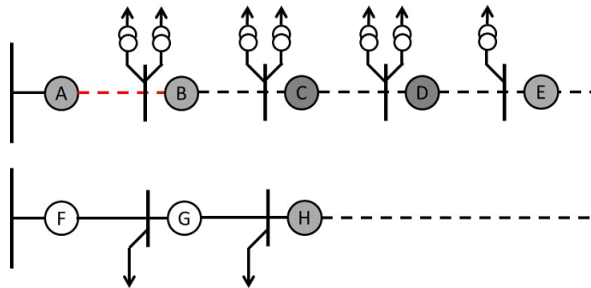


Figura 12 – Abertura das proteções que não possuem presença de tensão em ambos os lados.

Uma simulação computacional deste tipo de evento foi realizada utilizando o *framework* PADE com comunicação ponto a ponto entre os agentes. As ações realizadas pelos agentes podem ser observada pela janela de diálogos apresentada na Figura 14. Nesta figura, observa-se a



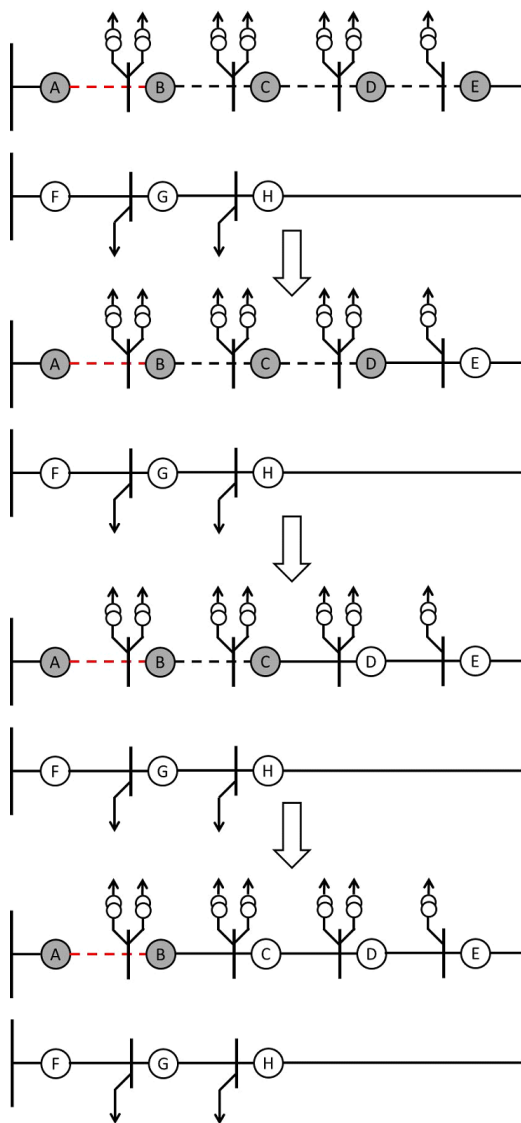


Figura 13 – Etapa de fechamento das chaves para recomposição do sistema após falta entre A e B.

interação dos agentes com o AMS do PADE, informando sua presença no sistema, identificadas pelas mensagens denominadas *Mensagem enviada ao agente*.

```
Run - my_pade
t_agents

[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHC@localhost:25664
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHA@localhost:9973
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHB@localhost:22943
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHB@localhost:60325
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHD@localhost:49039
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHG@localhost:16326
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHF@localhost:3396
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente Sniffer.Agent@localhost:60565
[AMS] 30/11/2018 13:56:32 --> Agente CHA@localhost:7055 identificado com sucesso
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHC@localhost:25664
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHD@localhost:49039
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHA@localhost:9973
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHB@localhost:60325
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente Sniffer.Agent@localhost:60565
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHF@localhost:3396
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHA@localhost:7055
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHB@localhost:22943
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHG@localhost:16326
[AMS] 30/11/2018 13:56:32 --> Agente X@localhost:57843 identificado com sucesso
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHD@localhost:49039
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHF@localhost:3396
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente X@localhost:57843
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHA@localhost:7055
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHG@localhost:16326
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHC@localhost:25664
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHB@localhost:22943
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente Sniffer.Agent@localhost:60565
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHA@localhost:9973
[AMS@localhost:8000] 30/11/2018 13:56:32 --> Mensagem enviada ao agente CHB@localhost:60325
[CHA@localhost:9973] 30/11/2018 13:56:32 --> Mensagem TRIP R.0 recebida
[CHA@localhost:9973] 30/11/2018 13:56:32 --> Falha detectada, chave CHA aberta
[CHB@localhost:60325] 30/11/2018 13:56:32 --> Nao ha presenca de tensao em ambos os lados do dispositivo, abrindo chave CHB
[X@localhost:57843] 30/11/2018 13:56:32 --> Mensagem INFORM Recebida
[CHC@localhost:60325] 30/11/2018 13:56:32 --> Setor em falta isolado!
[CHC@localhost:25664] 30/11/2018 13:56:32 --> Chave CHC fechada, sem presenca de tensao em ambos os lados, abrindo chave e repassando mensagem a jusante
[CHD@localhost:49039] 30/11/2018 13:56:32 --> Chave CHD fechada, sem presenca de tensao em ambos os lados, abrindo chave e repassando mensagem a jusante
[CHF@localhost:22943] 30/11/2018 13:56:32 --> Chave CHF fechada, sem presenca de tensao em ambos os lados, abrindo chave e repassando mensagem a jusante
[CHB@localhost:7055] 30/11/2018 13:56:32 --> Chave CHB aberta, mas possui presenca de tensao em um dos lados, requisitando informacoes do alimentador...
[CHB@localhost:7055] 30/11/2018 13:56:32 --> Alimentador possui 305 A disponivel, analisando possivel reconfiguracao da rede...
[CHB@localhost:7055] 30/11/2018 13:56:32 --> Fechando chave CHB
[CHB@localhost:22943] 30/11/2018 13:56:32 --> Restaurando setor, fechando chave CHD
[CHD@localhost:49039] 30/11/2018 13:56:32 --> Restaurando setor, fechando chave CHD
[CHC@localhost:25664] 30/11/2018 13:56:32 --> Restaurando setor, fechando chave CHC
[CHB@localhost:60325] 30/11/2018 13:56:32 --> Rede restaurada!
```

Figura 14 – Diálogo entre agentes obtidos para simulação computacional de falta no ramo entre os equipamentos A e B.

### 5.1.2 Falta após agente B:

O passo a passo para a restauração do sistema utilizando a comunicação ponto a ponto após falta no ramo entre os equipamentos B e C segue abaixo:

1. Proteção B atua isolando a falta (Figura 15).
2. Agente Switch B do equipamento que isolou a falta confirma estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ).
3. Agente Switch C, D e E mudam estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ), ao detectar a falta de presença de tensão por um tempo determinado (Figura 16).

4. Agente Switch B envia mensagem avisando que isolou a falta ao agente C.
5. Agente Switch C mantém sua chave aberta até ordem de entidade superior. Ademais informa ao agente D que falta foi isolada. Essa informação é passada a jusante até agente H.
6. Agentes D e E solicitam a jusante restauração de serviço informando carregamento pré falta do setor.
7. Agente H detecta presença de tensão em apenas um dos lados e realiza fechamento de chave perante atendimento de limite de carregamento adicional ao alimentador. Este processo se repete pelas ações dos agentes E e D nesta ordem. O processo de restauração é apresentado na Figura 17.
8. Fim do processo.

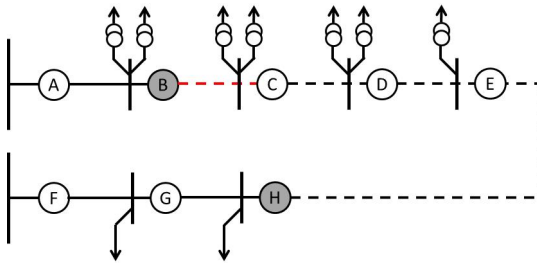


Figura 15 – Abertura da proteção B após ocorrência de falha entre B e C.

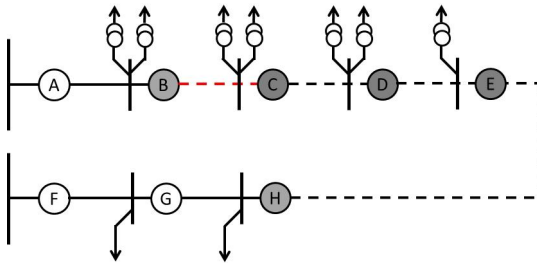


Figura 16 – Abertura das proteções que não possuem presença de tensão em ambos os lados.

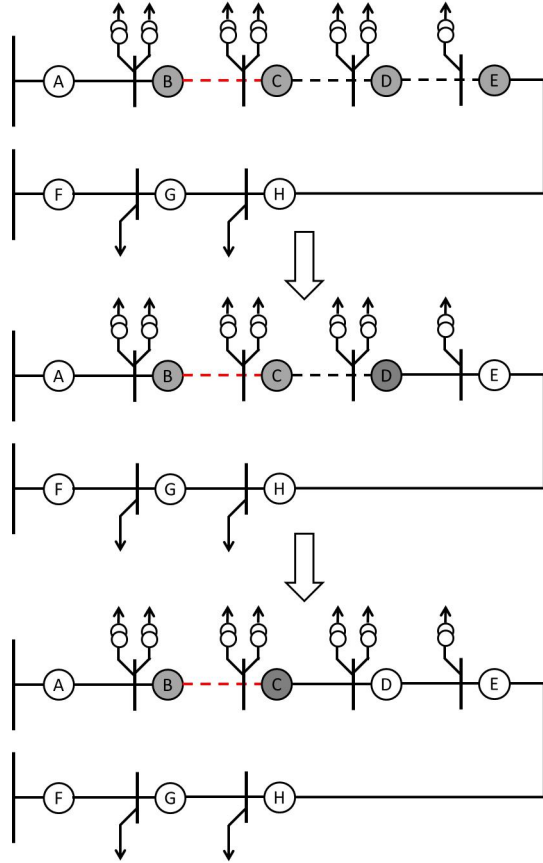


Figura 17 – Etapa de fechamento das chaves para recomposição do sistema após falta entre B e C.

### 5.1.3 Falta após agente C:

O passo a passo para a restauração do sistema utilizando a comunicação ponto a ponto após falta no ramo entre os equipamentos C e D segue abaixo:

1. Proteção C atua isolando a falta (Figura 18).
2. Agente Switch C do equipamento que isolou a falta confirma estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ).
3. Agente Switch D e E mudam estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ), ao detectar a falta de presença de tensão por um tempo determinado (Figura 19).
4. Agente Switch C envia mensagem avisando que isolou a falta ao agente D.
5. Agente Switch D mantém sua chave aberta até ordem de entidade superior. Ademais informa ao agente E que falta foi isolada. Essa informação é passada a jusante até agente H.
6. Agente E solicita a jusante restauração de serviço informando carregamento pré falta do setor.
7. Agente H detecta presença de tensão em apenas um dos lados e realiza fechamento de chave perante atendimento de limite de carregamento adicional ao alimentador. Este processo se repete pela ação do agente E. O processo de restauração é apresentado na Figura 20.
8. Fim do processo.

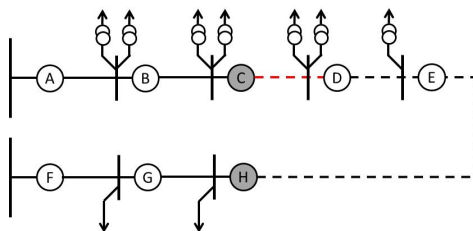


Figura 18 – Abertura da proteção C após ocorrência de falha entre C e D.

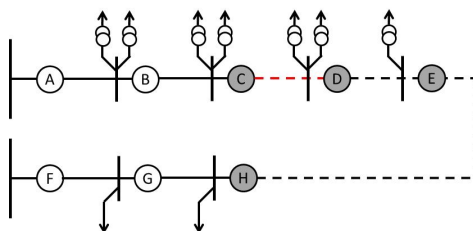


Figura 19 – Abertura das proteções que não possuem presença de tensão em ambos os lados.

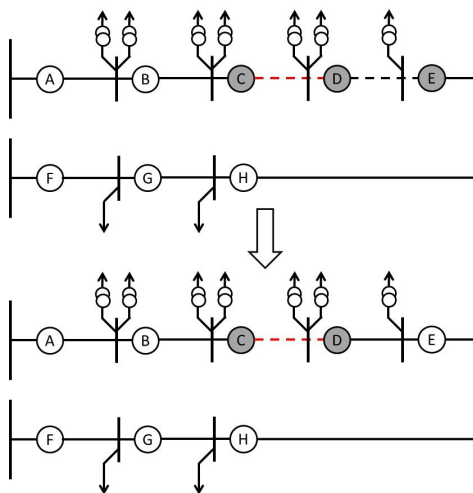


Figura 20 – Etapa de fechamento das chaves para recomposição do sistema após falta entre C e D.

### 5.1.4 Falta após agente D:

O passo a passo para a restauração do sistema utilizando a comunicação ponto a ponto após falta no ramo entre os equipamentos D e E segue abaixo:

1. Proteção D atua isolando a falta (Figura 21).
2. Agente Switch D do equipamento que isolou a falta confirma estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ).
3. Agente Switch E muda estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ), ao detectar a falta de presença de tensão por um tempo determinado (Figura 22).
4. Agente Switch D envia mensagem avisando que isolou a falta ao agente E.
5. Agente Switch E mantém sua chave aberta até ordem de entidade superior.
6. Fim de processo.

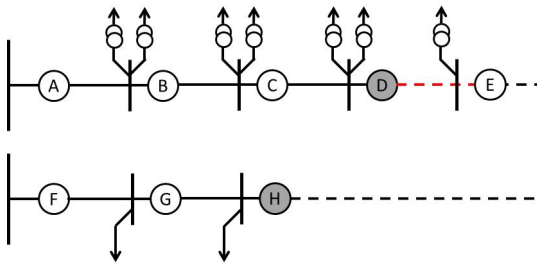


Figura 21 – Abertura da proteção D após ocorrência de falha entre D e E

Para a hipótese de falta no ramo a jusante da barra D foi realizado também uma simulação computacional, onde a caixa de diálogo dos agentes pode ser observada na Figura 23. Observa-se que com os passos realizados pelos agentes, a restauração não pode ser concluída com sucesso. Nota-se que após a abertura da chave D, envia-se uma mensagem a chave E, onde se verifica a não possibilidade de recomposição de cargas afetadas pela falta.

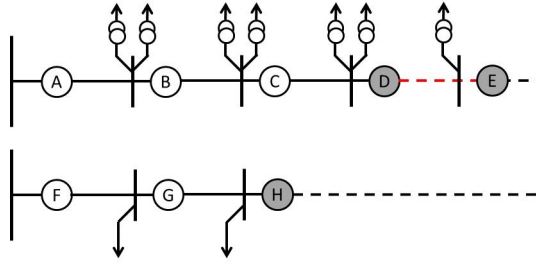


Figura 22 – Abertura das proteções que não possuem presença de tensão em ambos os lados

```
[CHD@localhost:10041] 30/08/2018 17:08:36 --> Mensagem TRIP R_0 recebida
[CHD@localhost:10041] 30/08/2018 17:08:36 --> Falha detectada, chave CHD aberta
[CHE@localhost:8339] 30/08/2018 17:08:36 --> Chave CHE ja encontra-se aberta, nao ha possibilidade de restauracao do trecho afetado
```

Figura 23 – Diálogo entre agentes obtidos para simulação computacional de falta no ramo entre os equipamentos D e E

### 5.1.5 Falta após agente F:

O passo a passo para a restauração do sistema utilizando a comunicação ponto a ponto após falta no ramo entre os equipamentos F e G segue abaixo:

1. Proteção F atua isolando a falta (Figura 24).
2. Agente Switch F do equipamento que isolou a falta confirma estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ).
3. Agente Switch G muda estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ), ao detectar a falta de presença de tensão por um tempo determinado (Figura 25).
4. Agente Switch F envia mensagem avisando que isolou a falta ao agente G.
5. Agente Switch G mantém sua chave aberta até ordem de entidade superior.
6. Agente H detecta presença de tensão em apenas um dos lados e realiza fechamento de chave perante atendimento de limite de carregamento adicional ao alimentador. O processo de restauração é apresentado na Figura 26.



7. Fim do processo.

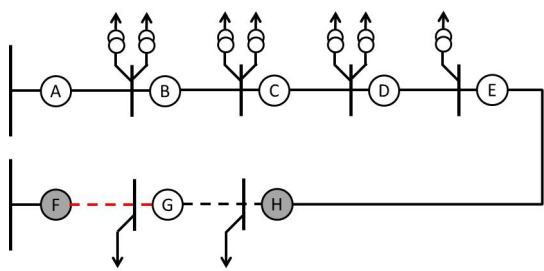


Figura 24 – Abertura da proteção F após ocorrência de falha entre F e G.

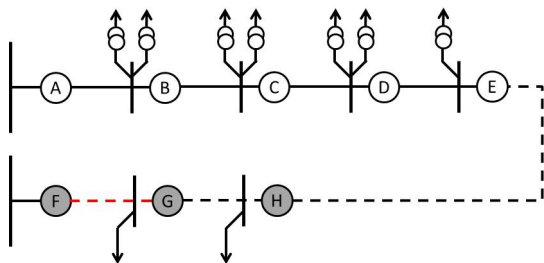


Figura 25 – Abertura das proteções que não possuem presença de tensão em ambos os lados.

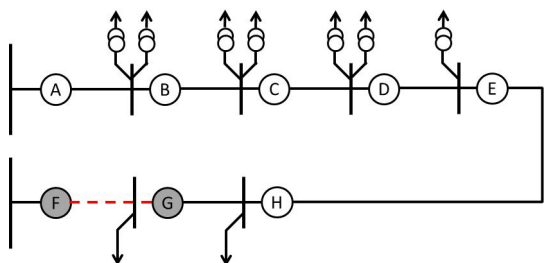


Figura 26 – Etapa de fechamento das chaves para recomposição do sistema após falta entre F e G.

### 5.1.6 Falta após agente G:

O passo a passo para a restauração do sistema utilizando a comunicação ponto a ponto após falta no ramo entre os equipamentos G e H segue abaixo:

1. Proteção G atua isolando a falta (Figura 29).
2. Agente Switch G do equipamento que isolou a falta confirma estado de posicionamento da chave de fechado ( $s = 1$ ) para aberto ( $s = 0$ ).
3. Agente Switch H continua em estado de posicionamento da chave em aberto ( $s = 0$ ).
4. Agente Switch G envia mensagem avisando que isolou a falta ao agente H.
5. Agente Switch H mantém sua chave aberta até ordem de entidade superior.
6. Fim de processo.

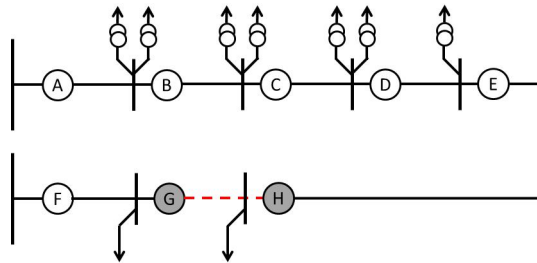


Figura 27 – Abertura da proteção G após ocorrência de falha entre G e H.

## 5.2 SÍNTESE DO CAPÍTULO

Neste capítulo foi apresentado simulações a partir da apresentação das possibilidades de recomposição do sistemas por meio do sistema RBTS. Observa-se que a utilização de agentes para recomposição

após falta segue passos pré estabelecidos em todas as hipóteses estudadas. As simulações realizadas se mostraram satisfatórias, em casos onde chaveamentos com outros alimentadores seja possível, grande parte do sistema pode ser recomposto sem que haja interferência externas nas ações.

## 6 CONSIDERAÇÕES FINAIS

### 6.1 CONCLUSÕES

O presente trabalho teve como objetivo desenvolver uma metodologia baseada em sistemas multiagentes para a recomposição automática de redes de média tensão, assim como verificar a aplicação do framework PADE para este problema.

Para a simulação de recomposição do sistema foi considerado que cada ponto de carga da rede RBTS possui um dispositivo embarcado com um Agente Switch. Cada agente possui como objetivos: monitorar as mudanças locais dos sinais de corrente e tensão, detectar a presença de falha no ponto, informar aos agentes vizinhos sobre a ocorrência de falta e realizar o processo de restauração após confirmação e análise de falta.

Os resultados obtidos nas simulações mostram que a reconfiguração automática de redes adotando a tecnologia de SMA é possível. Os agentes desenvolvidos foram capazes de encontrar uma solução e restaurar a rede após falta em casos onde havia a possibilidade de chaveamento com outro alimentador.

Finalmente, destaque-se que o framework PADE se mostrou adequado para a realização das simulações propostas no escopo desse trabalho. Entretanto, diversas limitações foram encontradas visando futuras aplicações, a saber: pouca documentação e exemplos disponíveis, não há uma representação explícita de ambientes para os agentes, também não existe a representação de arquiteturas para agentes (por exemplo, BDI - *Beliefs, Desires, Intentions*) e pouca comunidade desenvolvedora.

### 6.2 TRABALHOS FUTUROS

Como trabalhos futuros pretende-se:

- Implementar e melhorar a metodologia proposta em linguagem Jason [71];
- Desenvolver ambiente para os agentes utilizando o framework CArtAgO [72];
- Estudar modelos de organização de agentes para o problema;

- Simular e testar a proposta em rede real com elevado número de equipamentos.

## REFERÊNCIAS

- [1] R. E BROWN. *Electrical Power Distribution Reliability*. Boca Raton: CRC Press, 2009.
- [2] S Curcic, C.S Ozveren, L Crowe, and P.K.L Lo. Electric power distribution network restoration: a survey of papers and a review of the restoration problem. *Electric Power Systems Research*, 35(2):73 – 86, 1995.
- [3] Y. L. Lo, C. H. Wang, and C. N. Lu. A multi-agent based service restoration in distribution network with distributed generations. In *2009 15th International Conference on Intelligent System Applications to Power Systems*, pages 1–5, Nov 2009.
- [4] Y. T. Pan and M. S. Tsai. Development a bdi-based intelligent agent architecture for distribution systems restoration planning. In *2009 15th International Conference on Intelligent System Applications to Power Systems*, pages 1–6, Nov 2009.
- [5] I. H. Lim, M. S. Choi, S. J. Lee, and T. W. Kim. Intelligent distributed restoration by multi-agent system concept in das. In *2009 15th International Conference on Intelligent System Applications to Power Systems*, pages 1–6, Nov 2009.
- [6] I. S. Baxevanos and D. P. Labridis. Software agents situated in primary distribution networks: A cooperative system for fault and power restoration management. *IEEE Transactions on Power Delivery*, 22(4):2378–2385, Oct 2007.
- [7] J. M. Solanki, S. Khushalani, and N. N. Schulz. A multi-agent solution to distribution systems restoration. *IEEE Transactions on Power Systems*, 22(3):1026–1034, Aug 2007.
- [8] R. K. Mishra and K. S. Swarup. Smart distribution network restoration using multi agent system. In *2016 National Power Systems Conference (NPSC)*, pages 1–6, Dec 2016.
- [9] M. Elkhatib, M. Ahmed, R. Elshatshat, M. Salama, and K. B. Shaban. Distribution system restoration based on cooperative multi-agent approach. In *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*, pages 42–46, Sept 2015.

- [10] J. Ghorbani, S. Chouhan, M. A. Choudhry, and A. Feliachi. Hybrid multi agent approach for power distribution system restoration. In *2013 IEEE Energytech*, pages 1–5, May 2013.
- [11] N. Isherwood, M. S. Rahman, and A. M. T. Oo. Distribution feeder protection and reconfiguration using multi-agent approach. In *2017 Australasian Universities Power Engineering Conference (AUPEC)*, pages 1–6, Nov 2017.
- [12] M. J. Ghorbani, M. A. Choudhry, and A. Feliachi. A multiagent design for power distribution systems automation. *IEEE Transactions on Smart Grid*, 7(1):329–339, Jan 2016.
- [13] H. F. Habib and O. Mohammed. Decentralized multi-agent system for protection and the power restoration process in microgrids. In *2017 Ninth Annual IEEE Green Technologies Conference (Green-Tech)*, pages 358–364, March 2017.
- [14] J. V. C. Barros, R. F. Sampaio, R. P. S. Leão, L. S. Melo, and G. C. Barroso. A multiagent approach to automatic restoration of distribution network. In *2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*, pages 1–7, April 2013.
- [15] T. Kato, H. Kanamori, Y. Suzuoki, and T. Funabashi. Multi-agent based control and protection of power distributed system - protection scheme with simplified information utilization -. In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pages 49–54, Nov 2005.
- [16] A. L. Dimeas and N. D. Hatziargyriou. A mas architecture for microgrids control. In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pages 5 pp.–, Nov 2005.
- [17] I. H. Lim, S. J. Lee, M. S. Choi, and P. Crossley. Multi-agent system-based protection coordination of distribution feeders. In *2007 International Conference on Intelligent Systems Applications to Power Systems*, pages 1–6, Nov 2007.
- [18] E. M. Davidson and S. D. J. McArthur. Exploiting multi-agent system technology within an autonomous regional active network management system. In *2007 International Conference on Intelligent Systems Applications to Power Systems*, pages 1–6, Nov 2007.

- [19] T. Tsuji, T. Goda, K. Ikeda, and S. Tange. Autonomous decentralized voltage profile control of distribution network considering time-delay. In *2007 International Conference on Intelligent Systems Applications to Power Systems*, pages 1–6, Nov 2007.
- [20] A. L. Dimeas and N. D. Hatziargyriou. Control agents for real microgrids. In *2009 15th International Conference on Intelligent System Applications to Power Systems*, pages 1–5, Nov 2009.
- [21] E. M. Davidson, M. J. Dolan, S. D. J. McArthur, and G. W. Ault. The use of constraint programming for the autonomous management of power flows. In *2009 15th International Conference on Intelligent System Applications to Power Systems*, pages 1–7, Nov 2009.
- [22] Wan-Yu Yu, Von-Wun Soo, Men-Shen Tsai, and Yen-Bo Peng. Coordinating a society of switch agents for power distribution service restoration in a smart grid. In *2011 16th International Conference on Intelligent System Applications to Power Systems*, pages 1–7, Sept 2011.
- [23] A. L. Dimeas and N. D. Hatziargyriou. Operation of a multiagent system for microgrid control. *IEEE Transactions on Power Systems*, 20(3):1447–1455, Aug 2005.
- [24] M. M. Nordman and M. Lehtonen. Distributed agent-based state estimation for electrical distribution networks. *IEEE Transactions on Power Systems*, 20(2):652–658, May 2005.
- [25] M. M. Nordman and M. Lehtonen. An agent concept for managing electrical distribution networks. *IEEE Transactions on Power Delivery*, 20(2):696–703, April 2005.
- [26] I. S. Baxevanos and D. P. Labridis. Implementing multiagent systems technology for power distribution network control and protection management. *IEEE Transactions on Power Delivery*, 22(1):433–443, Jan 2007.
- [27] M. E. Baran and I. M. El-Markabi. A multiagent-based dispatching scheme for distributed generators for voltage support on distribution feeders. *IEEE Transactions on Power Systems*, 22(1):52–59, Feb 2007.



- [28] N. Perera, A. D. Rajapakse, and T. E. Buchholzer. Isolation of faults in distribution networks with distributed generators. *IEEE Transactions on Power Delivery*, 23(4):2347–2355, Oct 2008.
- [29] T. Xu, N.S. Wade, E.M. Davidson, P.C. Taylor, S.D.J. McArthur, and W.G. Garlick. Case-based reasoning for coordinated voltage control on distribution networks. *Electric Power Systems Research*, 81(12):2088 – 2098, 2011.
- [30] Aggelos S. Bouhouras, Georgios T. Andreou, and Dimitris P. Labridis. Feasibility study of the implementation of a.i. automation techniques in modern power distribution networks. *Electric Power Systems Research*, 80(5):495 – 505, 2010.
- [31] T. Logenthiran, Dipti Srinivasan, and Ashwin M. Khambadkone. Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system. *Electric Power Systems Research*, 81(1):138 – 148, 2011.
- [32] Mauro Rosa, Diego Issicaba, Wagner Franchin, and João Abel Pegas Lopes. Agent-based system applied to smart distribution grid operation. In Haiping Xu, editor, *Practical Applications of Agent-Based Technology*, chapter 1. InTech, Rijeka, 2012.
- [33] D. Issicaba, N. J. Gil, and J. A. P. Lopes. Islanding operation of active distribution grids using an agent-based architecture. In *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, pages 1–8, Oct 2010.
- [34] D. Issicaba, J. A. Pegas Lopes, and M. A. da Rosa. Adequacy and security evaluation of distribution systems with distributed generation. *IEEE Transactions on Power Systems*, 27(3):1681–1689, Aug 2012.
- [35] D. Issicaba, J. A. Pegas Lopes, and M. A. da Rosa. Distribution systems performance evaluation considering islanded operation. Aug 2011.
- [36] R. F. Sampaio, L. S. Melo, R. P. S. Leão, G. C. Barroso, and J. R. Bezerra. Automatic restoration system for power distribution networks based on multi-agent systems. *IET Generation, Transmission Distribution*, 11(2):475–484, 2017.

- [37] Yu Liu, Yunhe Hou, Shunbo Lei, and Dong Wang. A distribution network restoration decision support algorithm based on multi-agent system. In *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pages 33–37, Oct 2016.
- [38] X. Ji, L. Jian, X. Yan, and W. Hui. Research on self healing technology of smart distribution network based on multi agent system. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 6132–6137, May 2016.
- [39] Kunya Guo, Peng Jin, Liang Qi, and Qian Fu. A multi-fault rush repair strategy for distribution systems based on game theory of multi-agent. In *2015 Chinese Automation Congress (CAC)*, pages 1293–1298, Nov 2015.
- [40] A. Felix, H. S. V. S. K. Nunna, S. Doolla, and A. Shukla. Multi agent based restoration for smart distribution system with micro-grids. In *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 2341–2347, Sept 2015.
- [41] K. Anisha, M. Rathinakumar, N. Veerappan, and O. K. Satya Prakash. Multi agent based distribution system restoration with distributed generation. In *2014 IEEE National Conference on Emerging Trends In New Renewable Energy Sources And Energy Management (NCET NRES EM)*, pages 209–213, Dec 2014.
- [42] K. Anisha, M. Rathinakumar, and K. Prakash. Application of multi agent system for distribution system automation. In *2014 International Conference on Smart Electric Grid (ISEG)*, pages 1–7, Sept 2014.
- [43] F. de Oliveira Saraiva and E. N. Asada. Multi-agent systems applied to topological reconfiguration of smart power distribution systems. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 2812–2819, July 2014.
- [44] C. Silva and F. G. Guimarães. Reconfiguration of power distribution networks by multi-agent systems. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pages 546–550, Sept 2013.
- [45] X. Chen, B. Kong, F. Liu, X. Gong, and X. Shen. System service restoration of distribution network based on multi-agent technology. In *2013 Fourth International Conference on Digital Manufacturing Automation*, pages 1371–1374, June 2013.

- [46] A. Felix, H. S. V. S. Kumar Nunna, and S. Doolla. Distribution system restoration - a multi agent approach. In *2012 Annual IEEE India Conference (INDICON)*, pages 1014–1019, Dec 2012.
- [47] W. Khamphanchai, M. Pipattanasomporn, and S. Rahman. A multi-agent system for restoration of an electric power distribution network with local generation. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–8, July 2012.
- [48] G. Bodhinayake, L. H. P. N. Gunawardena, and K. T. M. U. Hemapala. Development of a multi agent system for voltage and outage monitoring. In *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pages 156–162, Feb 2017.
- [49] F. A. A. R. Marçal, R. P. S. Leão, J. B. Almada, R. F. Sampaio, F. L. M. Antunes, J. E. Rocha, and G. C. Barroso. Multi-agent systems applied for restoration of power networks supported by micro grids. In *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, pages 1–7, Nov 2016.
- [50] I. Kouveliotis-lysikatos, D. Koukoula, A. Dimeas, N. Hatziaargyriou, and S. Makriniakas. Overview of decentralized distribution system operation techniques. In *CIREN Workshop 2016*, pages 1–4, June 2016.
- [51] Z. Zhu, B. Duan, Z. Yang, Q. Yin, and J. Chen. A multi-agent and big data self-healing scheme for active distribution network with inverter-based distributed generators. In *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, pages 972–977, Nov 2016.
- [52] D. Shu, Z. Huang, J. Li, and X. Zou. Application of multi-agent particle swarm algorithm in distribution network reconfiguration. *Chinese Journal of Electronics*, 25(6):1179–1185, 2016.
- [53] A. Sharma, D. Srinivasan, and D. S. Kumar. A comparative analysis of centralized and decentralized multi-agent architecture for service restoration. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 311–318, July 2016.
- [54] Fei Ding and K. Loparo. Hierarchical decentralized network reconfiguration for smart distribution systems-part i: Problem formulation and algorithm development. In *2015 IEEE Power Energy Society General Meeting*, pages 1–1, July 2015.

- [55] M. A. Elgenedy, A. M. Massoud, and S. Ahmed. Smart grid self-healing: Functions, applications, and developments. In *2015 First Workshop on Smart Grid and Renewable Energy (SGRE)*, pages 1–6, March 2015.
- [56] A. Benigni, H. L. Ginn, A. Lowen, F. Ponci, and A. Monti. An embedded solution for multi-agent control of pebb based power electronic systems. In *2014 IEEE International Workshop on Intelligent Energy Systems (IWIES)*, pages 12–17, Oct 2014.
- [57] G. H. Merabet, M. Essaaidi, H. Talei, M. R. Abid, N. Khalil, M. Madkour, and D. Benhaddou. Applications of multi-agent systems in smart grids: A survey. In *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, pages 1088–1094, April 2014.
- [58] S. Fernández, F. J. Rodríguez, I. Sanz, C. Mataix, C. Girón, and M. Moranchel. Multi-layer agent-based architecture for smart grid monitoring. In *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 143–148, Oct 2013.
- [59] A. Sharma, W. A. Arokiasami, and D. Srinivasan. A multi-agent approach for service restoration with distributed generation. In *2013 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pages 1–6, Nov 2013.
- [60] M. E. Collins, R. W. Silversides, and T. C. Green. Control and coordination of a distribution network via decentralised decision making. In *2013 IEEE Power Energy Society General Meeting*, pages 1–5, July 2013.
- [61] W. Khamphanchai, M. Kuzlu, and M. Pipattanasomporn. A smart distribution transformer management with multi-agent technologies. In *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6, Feb 2013.
- [62] A. Zidan and E. F. El-Saadany. A cooperative multiagent framework for self-healing mechanisms in distribution systems. *IEEE Transactions on Smart Grid*, 3(3):1525–1539, Sept 2012.
- [63] H. Sekhavatmanesh and R. Cherkaoui. A multi-agent based analytical approach for service restoration in distribution networks. In *2017 IEEE Manchester PowerTech*, pages 1–6, June 2017.

- [64] Xuefeng Sun, Dongfei Lv, Haozhi Wang, Meiyi Hou, and Zhenxing Zhang. Topology identification method based on multi-agent for outage areas in multi-contact distribution network. In *2016 IEEE International Conference on Power and Renewable Energy (ICPRE)*, pages 456–459, Oct 2016.
- [65] A. Zidan, M. Khairalla, A. M. Abdrabou, T. Khalifa, K. Shaban, A. Abdrabou, R. El Shatshat, and A. M. Gaouda. Fault detection, isolation, and service restoration in distribution systems: State-of-the-art and future trends. *IEEE Transactions on Smart Grid*, 8(5):2170–2185, Sept 2017.
- [66] Michael Wooldridge. *An Introduction to MultiAgent Systems, 2nd Edition*, volume 30. 2009.
- [67] Fipa standard status specifications. <http://www.fipa.org/repository/standardspecs.html>, November 2018.
- [68] Lucas Silveira Melo. Desenvolvimento de uma plataforma para implementação de sistemas multiagentes com aplicação para recomposição automática de sistemas de distribuição de energia elétrica. Universidade Federal do Ceará, Dissertação de Mestrado, Fortaleza, 2015.
- [69] Roy Billinton, Sudhir Kumar, Nurul Chowdhury, Kelvin Chu, Kamal Debnath, Lalit Goel, Easin Khan, P. Kos, Ghavameddin Nourbakhsh, and J. Oteng-Adjei. A reliability test system for educational purposes-basic data. *IEEE Transactions on Power Systems*, 4(3):1238–1244, August 1989.
- [70] R. N. Allan, R. Billinton, I. Sjarief, L. Goel, and K. S. So. A reliability test system for educational purposes-basic distribution system data and results. *IEEE Transactions on Power Systems*, 6(2):813–820, May 1991.
- [71] Hübner-J.F. Bordini, R.H. Jason: A java-based agentspeak interpreter used with saci for multi-agent distribution over the net. <http://jason.sourceforge.net/>, 2005.
- [72] Alessandro Ricci, Mirko Viroli, and Andrea Omicini. Cartago: A framework for prototyping artifact-based environments in mas. <http://cartago.sourceforge.net/>, 2006.

## APÊNDICE A – Algoritmo do Agente Switch



---

```

from pade.misc.common import start_loop
from pade.misc.utility import display_message
from pade.core.agent import Agent
from pade.acl.messages import ACLMessage
from pade.acl.aid import AID
from pade.behaviours.protocols import FipaRequestProtocol
from pickle import loads, dumps

```

```

class CompRequest2(FipaRequestProtocol):

```

```

    def __init__(self, agent):
        super(CompRequest2, self).__init__(agent=agent,
            message=0, is_initiator=False)

    def handle_request(self, message):

        self.content = loads(message.content)

        if self.content['ref'] == 'R':
            if self.agent.data['v_up'] == 0 and self.agent.
                data['v_down'] == 0:
                display_message(self.agent.aid.name,
                                'Nao ha presenca de
                                tensao em ambos os
                                lados do dispositivo
                                , abrindo chave %s'
                                % self.agent.aid.
                                localname)
                self.agent.data['status'] = 0

        if self.content['ref'] == 'R_0':
            display_message(self.agent.aid.name, 'Mensagem
                TRIP R_0 recebida')

            resposta = message.create_reply()
            resposta.set_performative(ACLMessage.INFORM)
            resposta.set_content(dumps({'ref': 'R_0'}))
            self.agent.send(resposta)

        if self.content['dados']['load'] == 0:
            display_message(self.agent.aid.name, 'Falha
                detectada, chave %s aberta' % self.agent
                .aid.localname)
            self.agent.data['status'] = 'LOCKED'

        # se detectou falta , manda mensagem a
        #      jussante avisando sobre o isolamento da

```



```

        falta :
        message = ACLMessage(ACLMessage.REQUEST)
        message.set_protocol(ACLMessage.
            FIPA_REQUEST_PROTOCOL)
        message.set_content(dumps({'ref': 'R_01'}))
        for i in self.agent.data['down']:
            if i != '0':
                message.add_receiver(i)
        self.agent.send(message)

agentes = []
age = []
class CompRequest3(FipaRequestProtocol):
    def __init__(self, agent):
        super(CompRequest3, self).__init__(agent=agent,
            message=0, is_initiator=False)

    def handle_request(self, message):

        self.content = loads(message.content)

        if self.content['ref'] == 'R_01':
            display_message(self.agent.aid.name, 'Setor em
                falta isolado!')
            self.agent.data['status'] = 'LOCKED'
            data = {'id': self.agent.aid.localname, 'down':
                self.agent.data['down'], 'load': self.agent.
                data['load_limit'], 'status': self.agent.
                data['status']}

            #manda mensagem a jusante ate encontrar uma
            chave com possibilidade de restauracao,
            repassando os dados das chaves anteriores
            message = ACLMessage(ACLMessage.REQUEST)
            message.set_protocol(ACLMessage.
                FIPA_REQUEST_PROTOCOL)
            message.set_content(dumps({'ref': 'R_02', 'data'
                : data}))
            for i in self.agent.data['down']:
                if i != '0':
                    message.add_receiver(i)
            self.agent.send(message)

        if self.content['ref'] == 'R_02':

            if self.agent.data['status'] == 0:
                if self.agent.data['v_up'] == 0 and self.
                    agent.data['v_down'] == 0:

```

```

display_message(self.agent.aid.name, '
    Chave %s aberta, sem presenca de
    tensao em ambos os lados, repassando
    mensagem a jusante' %self.agent.aid
    .localname)
message = ACLMessage(ACLMessage.REQUEST)
message.set_protocol(ACLMessage.
    FIPA_REQUEST_PROTOCOL)
message.set_content(dumps({'ref': 'R_02'
    })))
for i in self.agent.data['down']:
    if i != '0' and i not in agentes:
        message.add_receiver(i)
        agentes.append(i)
self.agent.send(message)

else:
    if self.agent.data['status'] == 'LOCKED'
    :
        display_message(self.agent.aid.name,
            'Chave %s aberta, mas ramo
            encontra-se LOCKED_OPEN, nao ha
            possibilidade de restauracao!' %
            self.agent.aid.localname)

    else:

        display_message(self.agent.aid.name,
            'Chave %s aberta, mas possui
            presenca de tensao em um dos
            lados, requisitando informacoes
            do alimentador...' % self.agent.
            aid.localname)

        message = ACLMessage(ACLMessage.
            REQUEST)
        message.set_protocol(ACLMessage.
            FIPA_REQUEST_PROTOCOL)
        message.set_content(dumps({'ref': '
            R_03'})))
        for i in self.agent.data['up']:
            if i != '0':
                message.add_receiver(i)
        self.agent.send(message)

else:
    if self.agent.data['v_up'] == 0 and self.
        agent.data['v_down'] == 0:
        display_message(self.agent.aid.name, '
            Chave %s fechada, sem presenca de
            tensao em ambos os lados, abrindo
            chave e repassando mensagem a

```

```

        jusante' %self.agent.aid.localname)
self.agent.data['status'] = 'OPEN'
message = ACLMessage(ACLMessage.REQUEST)
message.set_protocol(ACLMessage.
    FIPA_REQUEST_PROTOCOL)
message.set_content(dumps({'ref': 'R_02'
    }))
for i in self.agent.data['down']:
    if i != '0' and i not in agentes:
        message.add_receiver(i)
        agentes.append(i)
self.agent.send(message)

if self.content['ref'] == 'R_03':

    if self.agent.data['up'] != 'FEEDER':
        if self.agent.data['status'] == 1:
            message = ACLMessage(ACLMessage.REQUEST)
            message.set_protocol(ACLMessage.
                FIPA_REQUEST_PROTOCOL)
            message.set_content(dumps({'ref': 'R_03'
                }))
            for i in self.agent.data['up']:
                if i != '0':
                    message.add_receiver(i)
                else:
                    pass
            self.agent.send(message)
        else:
            pass

    else:

        load_disp = self.agent.data['load_limit'] -
            self.agent.data['load']
        message = ACLMessage(ACLMessage.REQUEST)
        message.set_protocol(ACLMessage.
            FIPA_REQUEST_PROTOCOL)
        message.set_content(dumps({'ref': 'R_04', '
            load_disp': load_disp}))
        for i in self.agent.data['down']:
            if i != '0':
                message.add_receiver(i)
        self.agent.send(message)

if self.content['ref'] == 'R_04':
    if self.agent.data['status'] == 1:
        mess = loads(message.content)
        message = ACLMessage(ACLMessage.REQUEST)
        message.set_protocol(ACLMessage.
            FIPA_REQUEST_PROTOCOL)
        message.set_content(dumps(mess))

```

```

for i in self.agent.data['down']:
    if i != '0':
        message.add_receiver(i)
    self.agent.send(message)

else:
    if self.agent.data['status'] != 'LOCKED':

        display_message(self.agent.aid.name, '
            Alimentador possui %s A disponivel,
            analisando possivel reconfiguracao
            da rede....' %self.content['
            load_disp'])
        display_message(self.agent.aid.name, '
            Fechando chave %s' %self.agent.aid.
            localname)
        self.agent.data['status'] = 1
        self.agent.data['v_down'] = 1

        message = ACLMessage(ACLMessage.REQUEST)
        message.set_protocol(ACLMessage.
            FIPA_REQUEST_PROTOCOL)
        message.set_content(dumps({'ref': 'R_05'
            , 'load_disp': self.content['
            load_disp']}))
        for i in self.agent.data['down']:
            if i != '0':
                message.add_receiver(i)
                age.append(i)
        self.agent.send(message)

if self.content['ref'] == 'R_05':
    if self.agent.data['status'] != 'LOCKED' and
        self.agent.data['status'] == 'OPEN':
        if self.agent.data['load_limit'] <= self.
            content['load_disp']:
            display_message(self.agent.aid.name, '
                Restaurando setor, fechando chave %s
                ' %self.agent.aid.localname)
            self.agent.data['status'] = 1

            load_disp = self.content['load_disp'] -
                self.agent.data['load_limit']
            message = ACLMessage(ACLMessage.REQUEST)
            message.set_protocol(ACLMessage.
                FIPA_REQUEST_PROTOCOL)
            message.set_content(dumps({'ref': 'R_05'
                , 'load_disp': load_disp}))
            for i in self.agent.data['up']:
                if i != '0' and i not in age:
                    message.add_receiver(i)
                    age.append(i)

```

```

        for i in self.agent.data['down']:
            if i != '0' and i not in age:
                message.add_receiver(i)
                age.append(i)
        self.agent.send(message)

    if self.agent.data['status'] != 'LOCKED' and
       self.agent.data['status'] == 0:
        if self.agent.data['load_limit'] <= self.
            content['load_disp']:
            if self.agent.data['v_up'] == 0 and self
                .agent.data['v_down'] == 0:

                display_message(self.agent.aid.name,
                               'Fechando chave de
                               restauracao %s'
                               % self.agent.aid
                                   .localname)
                self.agent.data['status'] = 1

                load_disp = self.content['load_disp'
                    ] - self.agent.data['load_limit'
                    ]
                message = ACLMessage(ACLMessage.
                    REQUEST)
                message.set_protocol(ACLMessage.
                    FIPA_REQUEST_PROTOCOL)
                message.set_content(dumps({'ref': '
                    R_05', 'load_disp': load_disp}))
                for i in self.agent.data['up']:
                    if i != '0' and i not in age:
                        message.add_receiver(i)
                        age.append(i)
                for i in self.agent.data['down']:
                    if i != '0' and i not in age:
                        message.add_receiver(i)
                        age.append(i)
                self.agent.send(message)

    # else:
    #     display_message(self.agent.aid.name, '
    #         Alimentador nao possui carga suficiente
    #         para restaurar todos os setores, setor %
    #         s continua desernegizado, rede foi
    #         RESTAURADA PARCIALMENTE!' %self.agent.
    #             aid.localname)

    # else:
    #     #display_message(self.agent.aid.name, 'Rede
    #         restaurada!')

```

```

class AgenteSwitch(Agent):

    def __init__(self, aid, data):
        super(AgenteSwitch, self).__init__(aid=aid, debug=
            False)
        self.data = data
        self.fault_data = dict()

        # Comportamento que aguarda por ocorrencias na rede
        comp_request_2 = CompRequest2(self)
        self.behaviours.append(comp_request_2)

        # Comportamento de comando de IEDs
        comp_request_3 = CompRequest3(self)
        self.behaviours.append(comp_request_3)
        #
        # comp = CompRequest4(self)
        # self.behaviours.append(comp)

```

---